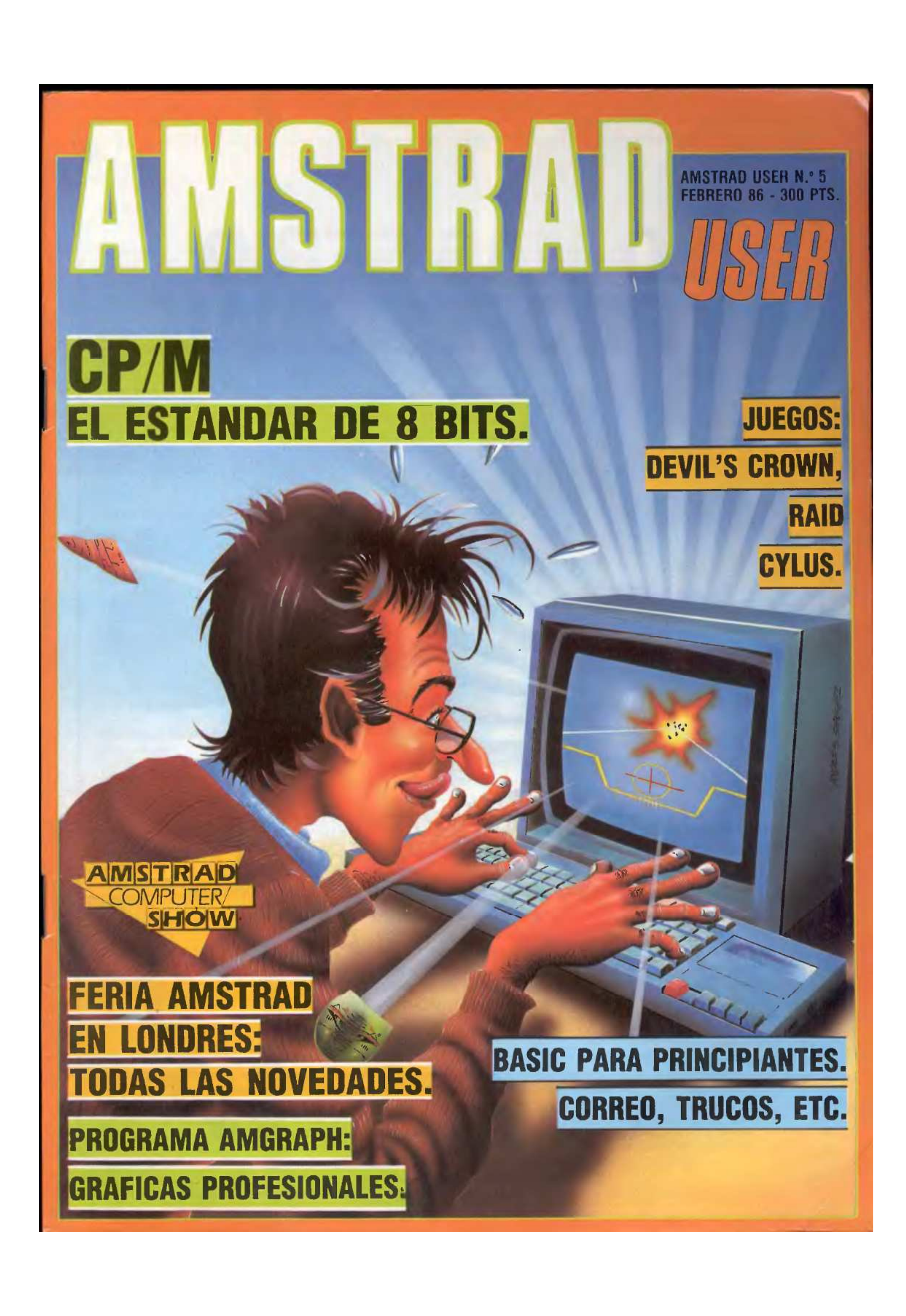


AMSTRAD
AMSTRAD USER N.º 5
FEBRERO 86 - 300 PTS.
USER
CP/M
EL ESTANDAR DE 8 BITS.
JUEGOS:
DEVIL'S CROWN,
RAID
CYLUS.
AMSTRAD
COMPUTER
SHOW
FERIA AMSTRAD
EN LONDRES:
TODAS LAS NOVEDADES.
BASIC PARA PRINCIPIANTES.
CORREO, TRUCOS, ETC.
PROGRAMA AMGRAPH:
GRAFICAS PROFESIONALES.





AMSTRAD-ATARI 520-COMMODORE SPECTRAVIDEO-SPECTRUM-QL

Cursillos de Basic y Servicio Especial Posventa





SOMOS PROFESIONALES

BRAVO MURILLO, 2 (aparc. gratuito en C/ Magallanes, 1) Tel. 446 62 31
DIEGO DE LEON, 25 (aparc. gratuito en C/ Núñez de Balboa, 114) Tel. 261 88 01 - MADRID
FELIPE II, 12 - 28009 MADRID



PRESENTANDO ESTE ANUNCIO TE HAREMOS
UN REGALO AL COMPRAR TU PCW 8256

Director
Santiago Gala

Subdirector
J. A. Sanz

Redacción
J. Ignacio Rey
Teresa Rubio
Justo Maurín
Ana M.ª Haro

Colaboradores
José A. Morales
Pedro Ruiz
Paco Suárez
Hugo Muñoz
Ian Hinton
Miguel Angel Barrios
Adolfo Martín Santos

Diseño
Enrique Ribas Lasso

Portada
Andrés Sánchez

Edita
Indescomp, S. A.

SERVICIO AL CLIENTE
Tel. (91) 433 44 58

Realización y Coordinación
Publinformática, S. A.

Jefe de Publicidad
Eloy Vergara

Dirección y Redacción
Bravo Murillo, 377, 5.° A
Tel. 733 74 13
28020 Madrid

Depósito legal
M-32038-1985

Distribuye
S.G.E.L.
Avda. Valdelaparra, s/n
Alcobendas (Madrid)

Fotocomposición
Amoretti
Sánchez Pacheco, 83
28002 Madrid

Fotomecánica
Karmat
Pantoja, 10
28002 Madrid

Imprime
Gráficas Velasco
Antonio Cabezón, 13
28034 Madrid



Editorial

La empresa *La Revoltosa* ha sido condenada a pagar un millón de pesetas a *La Casera* por uso de envases de esta última para sus productos, con la natural desorientación del cliente. Esta sentencia parece una prolongación lógica en España de la sentencia que condenaba a *Kodak* a abandonar toda su línea de productos de fotografía instantánea por violación de patentes de *Polaroid*, su máxima rival en dicho campo.

Aunque aparentemente el párrafo anterior guarda poca relación con Amstrad y sus usuarios, viene a colación porque la noticia es indicativa de una tendencia irrefrenable hacia una protección cada vez mayor de los derechos de propiedad «intangibles», sea de servicios o de uso. La entrada en el Mercado Común hará cada vez más difícil la supervivencia de las empresas (de hardware o de software) que no inviertan seriamente en productos de desarrallo propio. En el caso del software, una adecuada política de creación e importación se hace necesaria, si no queremos quedar prisioneros (para variar) de las compañías extranjeras.

El parque de Amstrad ha experimentado un gran crecimiento con la campaña navideña. Gran oportunidad, por tanto, para comenzar a escribir programas, ya que la demanda será cada vez mayor. Nosotros prometemos apoyo, siempre que la calidad les respalde, a todos los programadores que decidan trabajar para los Amstrad y para sus usuarios.

Por cierto, quizá cuando estas líneas lleguen a manos de los lectores los que siguen ahorrando, peseta a peseta, para poder comprarse su máquina se lleven una agradable sorpresa.

SUMARIO

AMSTRAD COMPUTER SHOW

NOVEDADES DE LONDRES

Eran las nueve de la mañana y, frente al hotel, se iba formando una cola cada vez más larga. Los usuarios de Amstrad aguardaban impacientes el comienzo, a las diez, del segundo Amstrad Computer Show, una feria en la que podrían contemplar y comprar a buen precio los últimos programas, periféricos y aparatos Amstrad. El sábado 11 de enero, a la diez y cinco, en el salón de exposiciones del Hotel Novotel no cabía un alma.

La expedición de Amstrad User había llegado el día anterior, aunque la niebla en Barajas provocó serios problemas. Paradójicamente, Londres nos recibió con un sol y un cielo despejado que parecían más propios de España que de las Islas Británicas.

La primera feria se celebró el pasado octubre, y fue un gran éxito, ya que apenas había acabado, y los organizadores ya tenía cubiertos los *stands* de la segunda convocatoria. En efecto, había grandes expectativas sobre las ventas de Amstrad en la campaña navideña, y muchos fabricantes y distribuidores de hard y software querían estar presentes con periféricos y programas en el mes de enero, para ser los primeros en llegar al público.

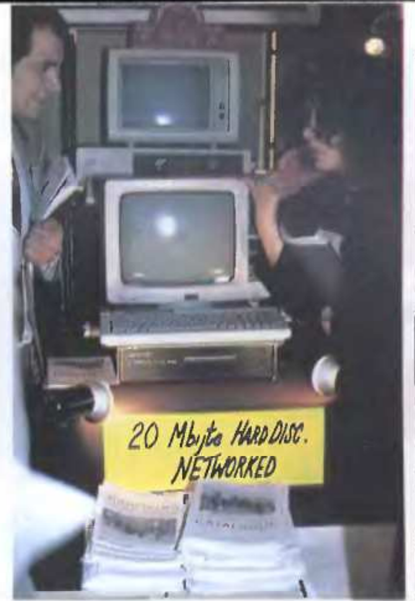


Más de 75 expositores, compañías de software y fabricantes y distribuidores de hardware, se repartían las amplias instalaciones del hotel. Entre las principales novedades, anunciadas poco antes en la prensa británica, llamaban especialmente la atención la ampliación de memoria anunciada por DK'tronics, que permitiría convertir un 464 o un 664 en un 6128, posibilitando el uso del CP/M Plus. La estrella de la feria era una placa que permite convertir al CPC 6128 en un compatible IBM PC, con todas las posibilidades que ofrece, al poderse ejecutar programas MS-DOS sobre las máquinas. Discos duros para 6128 y 8256 completaban las posibilidades de convertir a estas máquinas en verdaderos «monstruos», llenos de posibilidades en el terreno profesional.

Modems y dispositivos de comunicaciones eran otro de los focos de atención, y eran pocos los

stands donde no se ofrecía una contribución en ese terreno. Uno de los más ambiciosos era el puesto de **Northern Computers,** con una red local orientada al mercado educativo que ofrecía grandes posibilidades, al poderse conectar Amstrad, BBC, IBM PC, MSX y otras máquinas de red local, con un servidor de red equipado con disco duro de 20 megabytes. El sistema levantaba expectación, pero ésta se enfriaba un poco al ver los precios del sistema.

Los amantes de los gráficos y del diseño asistido por ordenador también estaban de enhorabuena, ya que tanto los diseñadores de software como los de hardware les dedicaban atención, con varios programas como The Animator, para presentar figuras en movimiento, o el Draughtsman, un excelente programa de CAD/CAM a la medida del usuario. Una tableta digitalizadora muy barata (unas 15.000 pesetas) teniendo en cuenta los precios habituales de estos dispositivos.



Una característica curiosa de la feria fue la presencia dominante de aplicaciones y material orientado directamente a aplicaciones «serias» (profesionales y educativas). Los Amstrad (especialmente el 8256, uno de los protagonistas de la feria) son máquinas que van a un público cada vez más profesional, aunque los programas de juegos estaban presentes en muchos de los *stands* de distribuidores de software, con una calidad excelente.

El PCW 8256 se compraba por docenas, sobre todo en uno de los expositores, que ofrecía la máquina con la segunda unidad (de 720K) a un precio poco mayor al del ordenador con un disco. El hermano mayor de la familia Amstrad había atraído la atención de muchas compañías, que transferían o escribían programas CP/M para usuarios profesionales.

Nuestra expedición recorrió arriba y abajo la feria, encontrando serias dificultades para abrirse paso entre la gente, y recogió kilos de catálogos y fotografías. También aprovechamos la ocasión para saludar a los editores de revistas para amstradictos ingleses y franceses; por cierto, estos últimos nos anunciaron su feria de novedades que se celebrará en París a partir del 22 de febrero, y donde se podrá seguir viendo qué productos de interés aparecen para complementar las prestaciones del ordenador.

LAS TARJETAS ELECTRONICAS: EL SOFTWARE DEL FUTURO

En el stand de Cumana pudimos ver la tarjeta ASTRON, una novedad muy interesante. Se trata de un interfaz que permite conectar tarjetas ROM, EPROM, EEROM y RAM permanente. Nos explicamos: mediante ellas, cualquier compañía de software puede comercializar, en grandes cantidades, programas de cualquier tipo, con el añadido de que, si se fabrican en grandes cantidades, su precio los hará poco atractivos de copiar.

Las restantes siglas significan que existen otras posibilidades de uso, como por ejemplo almacenar las fichas personales de un médico. En una EEROM la información es permanente, aunque se puede reprogramar. La versión de RAM con pilas admite hasta cinco años de memoria que se puede usar como un disco, pero mucho más rápida. Las capacidades oscilan entre 16 y 128K bytes.

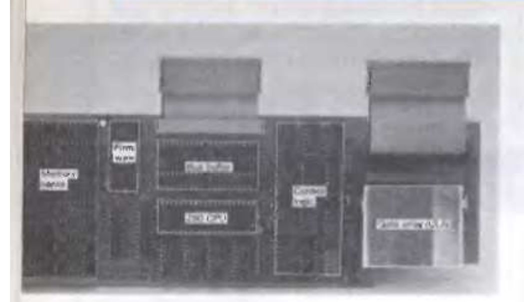


Los piratas, o se compran un dupicador de EPROM, o lo van a pasar realmente mal. (Incluso así, la copia no es tan fácil como puede parecer.)



AMSTRAD COMPUTER SHOW

VORTEX: UN VERDADERO REMOLINO VENIDO DE ALEMANIA



De Alemania Occidental viene Vortex, y su gama de periféricos para **Schneider** (nombre por el que se conoce allí al Amstrad) va a revolucionar el mercado profesional. Su primer producto anunciado, un disco duro de 20Mbytes, lo pudimos ver en la feria a un precio aproximado de 70.000 pesetas. Una interesante novedad para los usuarios profesionales, que pueden ahora obtener hasta 20 millones de caracteres de almacenamiento a velocidades más que interesantes.

Una ampliación de memoria de hasta 512K, con una complicada lógica de paginación, permite utilizar programas compuestos de varios bloques de hasta 32K cada uno, compartiendo variables y pudiendo saltar de un banco a otro mediante comandos incluidos en ROM. El resto de la RAM disponible se puede usar como buffer de impresora o disco virtual, o bien para almacenar pantallas.

Esta misma compañía anunció una tarjeta MS-DOS que debía haber sido la estrella de la feria. Sin embargo, puestos en contacto con Vortex, nos anunciaron que no estaba disponible por problemas de compatibilidad con su expansión de memoria. Nos anunciaron asimismo que el producto estará en funcionamiento en breve.

HISOFT: MUCHOS (BUENOS) PROGRAMAS

El catálogo de Hisoft no incluye sólo sus productos, que ya gozan de una muy buena reputación como los mejores ensambladores y compiladores de varios lenguajes para Spectrum y otras máquinas, sino que ha adquiridos los derechos de distribución de varias otras compañías que trabajan programas CP/M. El catálogo incluye el DEVPACBO, versión CP/M del más famoso conjunto de ensamblador y desemsamblador que haya existido en máquinas 280. Sus últimos desarrollos incluyen la posibilidad, en el caso del monitor MOM80, de interrumpir la ejecución en el momento que se desee, de buscar mnemónicos de ensamblador, de desemblar a disco,... El GEN80 en su última versión puede ser un macroensamblador que permite llamar macros recursivamente e incluir ficheros de disco.

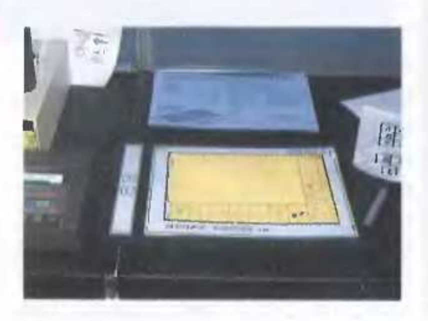
El nuevo catálogo permite elegir también el ensamblador M80 de Microsoft, el compilador C de Hisoft o el de BDS o AZTEC. Los compiladores e intérpretes de BASIC CBASIC, MALLARD y NEVADA hacen fácil elegir. También el FORTRAN y el COBOL NEVADA completan la lista de lenguajes, donde también Microsoft interviene con su MS-FORTRAN y su MS-COBOL, PRO-FORTRAN, Hisoft Pascal, Turbo Pascal y Pro Pascal hacen una lista de lenguajes donde resulta francamente difícil elegir. Forth, Modula 2, Micro-Prolog y PILOT...

Hará falta todo un artículo, que pensamos publicar en breve, para analizar las alternativas de todos estos lenguajes. Mientras tanto, habrá que fiarse de la accesibilidad de los paquetes de software y de la compatibilidad con desarrollos anteriores.

TABLETA DIGITALIZADORA A UN PRECIO ASEQUIBLE

La tableta digitalizadora Grafpad II, que permite el dibujo a mano y la introducción de datos, es la primera que cae (en Gran Bretaña) por debajo del límite de las 15.000 pesetas. Algunos lectores se extrañarán, pero hace un año era imposible encontrar una tableta digitalizadora por menos de 50.000 pesetas.

Con una resolución de 1.280 × 1.024 puntos, una velocidad muy interesante y un interfaz paralelo adaptado a BBC, Commodore y Amstrad, se trata de un instrumento útil para dibujantes, diseñadores y gente interesada en CAD/CAM. El programa de dibujo que se incluye permite (y lo vimos hacer al público) dibujar con casi tanta facilidad como con lápiz y papel.



DK'TRONICS: NOVEDADES HARDWARE

Una pila de expansiones es lo que ha preparado los chicos de dk'tronics. Una expansión de memoria de 64K y otra de 256 permiten la conversión de un 464 o de un 664 (al menos eso nos dijero) en una máquina tan potente como un CPC 6128, e incluso con más memoria, que puede usar el CP/M Plus (si se dispone de una licencia de uso).

Los posibles usos de la memoria extra, aparte de la mejora de poder usar CP/M Plus, con un área de programas de usuario de 61K, vienén dadas por software de conmutación de bancos análogos al bankman del 6128, para usar como medio de lamacenamiento temporal. Los precios en Londres oscilaban entre las 12.000 de la expansión de 64K y las 24.000 de la de 256K.

Para los usuarios que tengan más problemas con el almacenamiento en disco, se ofrece un «disco de silicio» de 256K, con software que le permite emular inmediatamente la unidad B (la C en sistemas con dos discos), para ganar tiempo en las operaciones de disco. El precio es el mismo que el de la ampliación de 256K.



Otras novedades de interés que amplian la oferta de periféricos ya existentes son el lápiz óptico y el sintetizador de voz, este último con un amplificador estéreo.

NOVEDADES DE DIGITAL RESEARCH

Digital Research no sólo inventó y comercializó el CP/M, sino que dispone de un gran catálogo de productos para ese operativo. Entre ellos podemos reseñar la aparición de su CBASIC y del Pascal MT+. Estos dos programas, compilador de BASIC y Pascal, ayudan a establecer la gama Amstrad como una de las mejores dotadas de herramientas de desarrollo de programas.

Por otra parte, DR Graph es uno de los programas de gráficas para el profesional más interesantes de la gama CP/M, y DR Draw un programa de dibujo de altas prestaciones.

Los cuatro programas forman parte de la gama profesional de Digital Research, y se presentan con una presentación impecable y a un precio asequible en Gran Bretaña.



MODDEM Y CONEXIONES CON SISTEMAS DE TELETEXT

Entre los modem que se ofrecian, llamaba la atención el equipo de Cirkit, formado por un Interface específico para los Amstrad y el acoplador acústico de Protek. Orientado a la comunicación con servicios de videotexto, su principal ventaja es el precio, de alrededor de 7.000 pesetas. Su inconveniente, que sólo soporta tasas de comunicación de 1200/74 y 1200/1200 half duplex.



Los programas incluidos servían para la comunicación y transferencia de ficheros entre usuarios y con los boletines electrónicos.

El modem Nightingale, comercializado por PACE, ofrecía unas prestaciones mucho mayores, pero a un precio de 25.000 pesetas: se trata de un modem multiestándar (300/300, 1200/75 y 1200/1200), con modos orígenes y respuesta, que se acopla electrónicamente al teléfono y vía un interface RS232 proporciona al Amstrad amplias posibilidades de comunicación. El software incluido cubre un gran número de posibilidades.

La clave

Bien, hasta ahora hemos contado la historia anterior a la presentación del CPC464. Con esta última entrega les relataremos los sucedido desde entonces hasta la fecha.

Después de la excitación que produjo el lanzamiento del 464 en abril de 1984, el equipo de desarrollo se concentró en terminar el sistema de disco —que apareció puntualmente en las tiendas en octubre de ese mismo año—. Al tiempo que se desarrollaba el trabajo en torno a 464, la industria parecía apasionada con los cassettes —aunque el jefe se dio cuenta rápidamente de que los discos iban a ser elementos punteros, al comparar los diez segundos que llevó cargar desde disco el programa «La Pulga», comparado con los seis minutos que tardaba en cassette—.

mecanismo que ofrecía alguna esperanza de que Amstrad consiguiera un precio mágico de 199 libras fue una oferta de Panasonic. La saga de los discos de tres pulgadas había comenzado.

El hecho de que las unidades de tres pulgadas ofrecían mucha más seguridad que las configuraciones más económicas de 5 pulgadas fue también un factor importante en la toma de decisión. En ese tiempo, los discos de 3 1/2 pulgadas no estaban del todo reconocidos, y esa fragilidad en cuanto a posición de mercado fue un motivo de preocupación.

Pero no es estándar

Pero el precio no fue el único problema. Para un sistema de disco completo con sistema operativo y con compatibilidad total con el sistema de cassette, un precio de 199 libras era un auténtico bombazo. Se emplearon muchas horas reflexionando sobre la oportunidad de introducir discos de tres pulgadas en un mercado dominado por las unidades de 8, 5 1/4 y 3 1/2, y había tantos formatos diferentes dentro de este espectro que el uso de la palabra «estándar» carecía de sentido.

Sin embargo, los discos de tres pulgadas tenían la ventaja de que eran eléctricamente idénticos al estándar de 5 1/4, aunque eso era lo más que podía decirse en ese momento para inquietar al mercado de tres pulgadas y media.

Algunos observadores influyentes dieron muestras de un alto grado de ignorancia comparando solamente capacidades. Manifestaban entonces que los discos de tres pulgadas tenían una capacidad de 250 K, mientras que los de tres y media ofrecían hasta 1 Mbyte. Bien, lectores, esa era la situación de todas las alternativas, pero los precios se habían situado fuera de los términos de referencia de Amstrad para un productos de masas. Con el PCW8256 y la erosión de costes que se produjo, todo esto salta a la luz, aunque no hay evidencia de que el profeta de las tres pulgadas y media se haya comido sus palabras.

¿Quién es el estándar ahora?

Amstrad ha vendido probablemente más sistemas CP/M-80 que cualquier otro fabricante de micros, y por eso debe ser considerado como un estándar. Dando otros pocos meses de ventas a los enormemente exitosos PCW8256, CPC6128 y a la expansión DD1 no habrá más argumentos en contra.

Teniendo en cuenta la reacción hacia un micro doméstico con un sistema de disco de bajo costo, era obvio que lo que había que hacer era incluir el disco dentro del ordenador, y extender el principio básico de Amstrad de reducir costes poniéndolo todo en un solo módulo.

Era, supuestamente, una tarea para llevar a cabo con el mínimo de interrupciones, ya que el PCW8256 se había concebido en ese tiempo, y era el foco de mayor atención. Sin embargo, Locomotive vio la oportunidad de adaptar el Basic al sistema de disco y de añadir al mismo tiempo algunas características extras que no se incluyeron en la versión 1.0 por falta de tiempo.

El 6128 fue considerado, desde el principio, como un imposible, pero MEJ encontró eventualmente una manera de solucionar los problemas que se plantearon al intentar añadir 64 K de RAM —Locomotive encontró la manera de hacer uso de ello— y el proyecto se completó más rápidamente de lo que nadie había esperado. El precio de las partes se vino abajo rápidamente como resultado del agitamiento que se produjo a principios del 85 con el sorprendente precio del 6128 (299 libras).

Software

Mientras tanto, el software empezó a ser abundante. La tarea original de convencer a las casas de soft para que probaran nuestros programas se hizo más fácil durante ese tiempo, y algunas marcas con renombre comenzaron a aparecer. «Roland in Time» fue el primer mega-juego, perseguido muy de cerca por la conversión de «Jet Set Willy»

el precio



(¿No os parece todo como de una generación anterior?).

La abundante documentación del sistema operativo fue un acicate para muchos programadores. El Pascal de HiSoft estuvo disponible rápidamente para demostrar lo que se podía hacer con la máquina y un sistema operativo cuando caía en manos de un programador serio.

La compañía Tasman evaluó rápidamente el potencial del sistema y produjo, en la forma de Amsword, la mejor implementación de su paquete TasWord Spectrum WP, animando a algunos especialistas a tratarlo con grandes elogios y también con algo de entusiasmo. De hecho, el éxito del AmsWord subrayó la relevancia de un procesador de textos para este tipo de domésticos y tuvo bastante influencia en la dirección de nuevos productos.

Las compañías de software que originalmente no aceptaron la oferta de Amstrad empezaron a interesarse al tiempo que leían los informes de ventas, de exportación y situación general de la empresa. Con BBC y Sinclair, cuyos mercados mostraban el comienzo de los problemas que siguieron durante 1985, un buen número de casas de software estaban ahora tratando de mantener íntegra su posición en el mercado.

Cada vez iba apareciendo más y mejor software, y a mediados del 85 todos los grandes estaban produciendo nuevos títulos simultáneamente en el formato del 464.

Te lo dije...

Dando muestras de una hospitalidad fuera de lo común, Amsoft organizó una conferencia para escritores de software en diciembre del 84 para estimular y para proporcionar información general, insinuando que el esfuerzo que se hiciera en el desarrollo software para CP/M no quedaría sin recompensa.

A pesar de eso, la introducción del 6128 y del 8256 parecía haber cogido a la mayoría de la gente en el limbo, y el innovador software de CP/M con el soporte gráfico de GSX proviene ahora de uno o dos sitios solamente —sobre todo de Ian Seagle de Software Technology—.

Parece incuestionable la fe que la gente de Amstrad tiene para con su producto. La ruptura y debacle general del mercado de micros domésticos provoca, en aquellos que no están próximos al producto, suspicacias y una actitud cauta si se pretende involucrarlos activamente. Quizás el dinamismo de Amstrad en mercados volátiles hace que la gente se muestre cautelosa de que Amstrad pudiera desaparecer cuando las cosas fueran mal, pero lo que ellos no aprecian correctamente es que la calidad de la oposición en el mercado de ordenadores no era tan alta como en el mercado electrónico, que forma la columna de la experiencia de Amstrad.

Para Amstrad, competir fue como pescar en un barril. La competencia ayudó generosamente al esfuerzo general de la compañía con sus continuos tropiezos, que se traducían en precios erróneos, tácticas de crecimiento lento y una falta general de perspicacia en los negocios que fue el resultado de dos años previos de adulación por parte de los medios de difusión.

Contrariamente a la creencia popular, ni una institución ni el amparo de la BBC fueron garantía de invencibilidad una vez que el muchacho de Hackney olfateó el triunfo. Este hecho produjo algo de confusión y sorpresa en el mercado de valores, lo cual vino a demostrar que todas las compañías fabricantes no estaban en un buen momento. No eran

tiempos de ofrecer grandes dividendos, así que Amstrad estableció los suyos alrededor de los 85 peniques hasta hace muy poco.

Sin embargo, todo ello significa que Amstrad parece capaz de producir nuevos ordenadores antes de que Ian Sinclair pudiera escribir sobre ellos, y eso quiere decir algo.

De nuevo en el hardware

El 6128 fue, preferentemente, una labor de fondo, mientras que los mayores esfuerzos se están empleando en el PCW8256. Este comenzó con la pantalla al final, pero por razones de diseño y tecnología, se eligió una postura más convencional. A pesar de todo, el principio básico de meter todo lo posible en el módulo de la pantalla se mantuvo, y las grandes ventajas de su formato de 90 por 32 aún provocan las quejas de los comercializadores de PCs, ya que el público acepta gozosamente la oferta de Amstrad en su justo valor y utilidad, en lugar de los compatibles IBM, que son costosos y poco potentes, y tienen pantallas con una capacidad de un 40 % menor.

El colapso de precios de los componentes provocó la adopción de técnicas «todo RAM», y aunque se había considerado como posibles otras opciones en caso de que la RAM de 256 K no bajara de precio de forma ostensible, no fueron necesarias. En consecuencia, los 112 K de disco RAM es una de las características más útiles del sistema, ya que permite que los programas en CP/M simplemente «estén» y no se gasta tiempo en procesos de carga adicionales.

MEJ Electronics empleó la mayor parte de su tiempo desarrollando las matrices de puertas lógicas y, una vez más, en realizar un planning de trabajo. En vista de la manera en que se desarrollaba el software para controlar el hardware, era esencial para Locomotive tener una ejemplo práctico tan pronto como fuera posible, de forma que se desarrolló un complejo sistema de simulación, más aún que el del 464, ya que esta vez había que utilizar menos circuitos integrados, y no había componentes como el controlador de pantalla 6845. LocoScript pasó varios cientos de fases de desarrollo. Como se trataba de un producto complejo y totalmente integrado, cualquier alteración, por pequeña que fuera, requería una recopilación completa del código fuente. Pero con su usual atención a los detalles y a la calidad, la versión 1.0 se hizo realidad y demostró ser mucho más sofisticada y capaz de lo que nadie hubiera esperado, dado el plazo de tiempo relativamente corto de que se dispuso.

Y en caso de que usted se lo pregunte, LocoScript no fue el primer nombre elegido, ya que se desecharon muchos otros igualmente buenos y descriptivos. Piense usted que WordStar tampoco fue muy original, ¿no es cierto?



NOTICIAS

Problemas, problemas, problemas...

Excepto Amstrad, los fabricantes de microordenadores siguen empantanados en una grave crisis: Commodore cierra la planta británica, única donde se fabricaba el Commodore 16, para concentrar sus esfuerzos sobre el resto de las factorías. Y, por otro lado, Sir Clive Sinclair ha acabado el año en números rojos.

Los malos resultados en el mercado, junto a la bajada de la valoración de stocks (sobre todo de QL), han hecho que Sinclair pase de unos beneficios de 14 millones de libras a unas pérdidas de 18 millones para el ejercicio que acabó el 31 de marzo de 1985. Si la valoración de los stocks no hubiera cambiado, Sinclair habría ganado cuatro millones de libras. Se han planteado serias reestructuraciones en la empresa, que le han permitido sobrevivir cuando muchos auguraban lo peor.

Por otro lado, la empresa sigue anunciando novedades. Al Sinclair Enigma, que saldrá con 1 Mbyte de memoria y el procesador 68000 de Motorola, se le añade el 128 K, para el que se anuncia la versión británica con disco de tres pulgadas y media y CP/M. Más usuarios CP/M, lo que redundará en programas mejores y más baratos para los amstradictos con disco.

A las duras..., y a las maduras

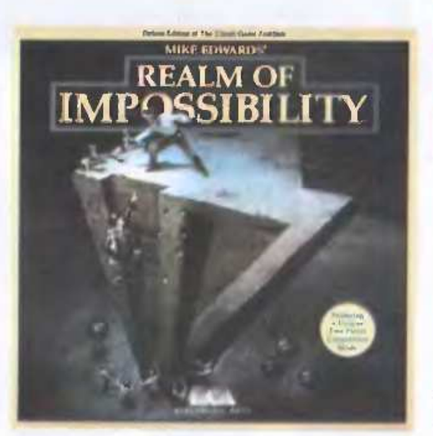
La *movida* la anunciamos en nuestro número de enero: Mastertronic sacaba programas para Amstrad a un precio inigualable. El resultado en las listas de éxitos británicos fue espectacular, con cinco programas entre los diez primeros en la lista para Amstrad.

Ahora, muy poco después, aparece la noticia de que, en la prensa británica, la hermandad de compañías de *software* piden que se excluyan los productos «de rebajas» de las listas o, al menos, que se hagan dos listas separadas, una para los productos «de precio completo» y otra para las «rebajas». Mastertronics, naturalmente, ha protestado y la solución no se ha tomado todavía. La hermandad de fabricantes dice que, aunque los más vendidos se indican por la cantidad, eso no da a los usuarios una idea de la calidad final del juego, o incluso del volumen del negocio, ya el precio es mucho menor.

El portavoz de Mastertronics dijo que no creía que tuviera sentido vender juegos a 2.500 pesetas los usuarios de máquinas que valen menos de 60.000. «Los precios de los juegos tienen que caer, ya que los precios de las máquinas caen. Las demás casas de software van a perder la batalla. Antes de incluir los juegos de Mastertronic en la lista, nadie creía que pudiéramos tener éxito, pero este año hemos vendido más de dos millones de juegos».

Por otro lado, Gallup, que es la asociación que se encarga de elaborar las listas, escucha a ambos contendientes, y espera tomar en breve una decisión. Nosotros esperamos que España sea en breve objeto de batallas parecidas, que pongan los precios de los programas a un nivel que acabe con los piratas por asfixia económica.

Electronics Arts busca los mejores juegos en Gran Bretaña



Uno de los fundadores de Electronic Arts, una empresa de los EE.UU. que ha comercializado algunos de los mejores juegos de Commodore y tiene un plan de lanzamientos importante para Amstrad (ya hemos hablado de su distribución en España por DRO soft), está por Gran Bretaña buscando juegos interesantes para el mercado norteamericano.

Según sus propias declaraciones, llevan ya suficiente tiempo vendiendo juegos *made in USA* en este lado del Atlántico, y ya es hora de que los buenos escritores europeos tengan oportunidad de acceder al mercado norteamericano. A más de un autor español le brillarán los ojos ante las posibilidades que le abriría una entrevista con Tim Mott, que así se llama el «cazador de talentos».

Un robot controlado por ordenador

Se trata de una máquina educativa, parte de la línea de productos Fischertechnik Computing. Esta empresa, especializada en cajas de construcción, pero con una gran experiencia en el campo industrial, decidió entrar en el campo de la informática personal ofreciendo *kits* de montaje para usuarios que quieran construir sus propios periféricos.

El *kit* incluye tres elementos: en primer lugar el interfaz, que permite su conexión a Amstrad 464, 664 ó 6128, Commodore o Apple, de manera que el ordenador pueda controlar el robot de una forma efectiva. El equipo de construcción propiamente dicho sirve para construir hasta diez montajes, sin contar con la imaginación del usuario. Un rotor de antena, un semáforo, elevador de cargas, un *plotter* y un sistema de introducción gráfica son sólo una parte de sus posibilidades, con un panel solar, un sistema de ordenaciónde piezas por tamaño y un robot aprendiz como realizaciones máximas.

El sistema se complementa con el software que permite su manejo desde el ordenador, que, en el caso de Amstrad, es accesible en cassette o diskette.



La distribución en España es de Master Computer, y el precio está a la altura de las prestaciones: alrededor de 35.000 pesetas.

NOTICIAS



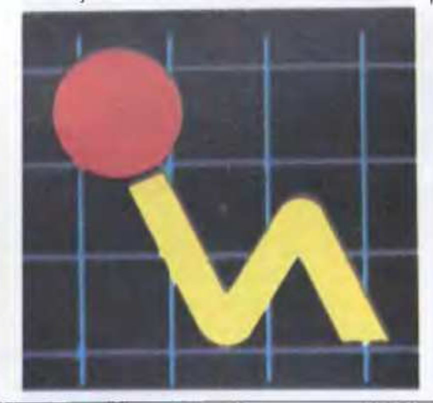
Nuevos periféricos de MHT Ingenieros



La empresa MHT Ingenieros, conocida por los periféricos realizados para Spectrum y Amstrad, ha sacado recientemente un interfaz RS232 para Amstrad, que permite explotar al máximo la potencia de este interfaz de comunicaciones. La unidad se acopla fácilmente al Bus de expansión de los CPC, y se puede controlar ampliamente a través de comandos RSX en ROM. Admite velocidades diferentes en recepción y transmisión, por lo que resulta muy apropiado para el uso como conector para un modem de comunicaciones.

Otros proyectos, alguno en muy avanzado estado de desarrollo, son una ampliación de memoria de 64K, de uso general como segundo banco de memoria para 464, 664 y 6128, que se podrá usar también como memoria intermedia de impresora, ahorrando tiempo en la impresión de listados largos. Un sintetizador de voz con software orientado al idioma castellano ahorrará los malabarismos que deben realizar ahora los usuarios de este producto, con programas de comunicaciones pensados para el idioma inglés.

A más largo plazo, sus proyectos incluyen el desarrollo de productos que permitan la comunicación entre ordenadores, así como una ampliación de memoria de hasta 256K, que podrá utilizarse en múltiples aplicaciones aunque todavía no se sabe si servirá para que los usuarios de 664 y 464 con disco puedan ejecutar CP/M Plus.



II Convención Indescomp

Por segunda vez, Indescomp realiza una convención para presentar a distribuidores, proveedores, prensa y público en general sus resultados del año 1985, así como para lanzar su nueva estrategia de ventas para 1986. Una reunión que se celebrará apenas haya salido de imprenta este número, el día 7 de febrero, y de la que esperamos dar información en nuestro número siguiente.

Los resultados del años muestran un desbordamiento total de las previsiones de ventas realizadas en octubre, hace menos de cuatro meses, cuando se presentó el nuevo 8256.

CP/M

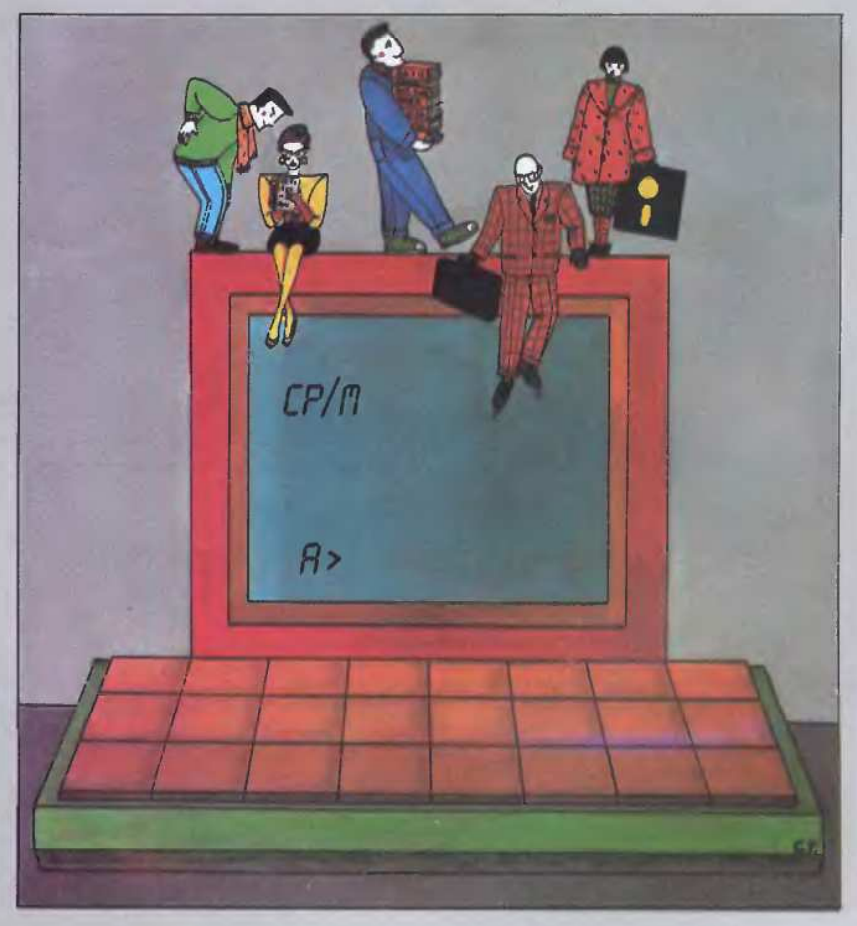
PARA CONTROLAR TOTALMENTE LA MAQUINA

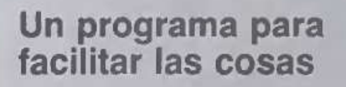
El Sistema Operativo es el alma que le da vida a la fría máquina. Y, aunque esta frase resulte sabida por todo el mundo, se olvida con frecuencia, cuando nos referimos al BASIC y al Sistema Operativo, que ambos no son más que dos programas.

El CP/M, sistema operativo que presentan los Amstrad, es un programa muy especial, ya que se encarga de la difícil tarea de coordinar todas las tareas del ordenador.

El usuario se sienta ante su máquina, introduce el disco con el operativo, aguarda un momento y ¡hop!, aparece un mensaje de conexión. Suele incluir el nombre y versión del CP/M (2,2 o Plus), y el fabricante o distribuidor (en este caso Amstrad). A continuación aparece el número de discos disponibles, si hay o no puerto serie, etc., y la memoria libre. A continuación la máquina nos indica con el famoso A> que espera que introduzcamos alguna orden.

Este famoso mensaje se suele llamar *prompt*, e indica que la máquina espera una orden. La máquina ejecuta en ese momento un programa llamado CCP (Console Command Processor) o Intérprete de Comandos del teclado. Su trabajo es sencillo, aunque aburrido: se trata de leer una línea del teclado, ejecutar sus instrucciones y volver a leer otra. El CCP, o intérprete de comandos, forma parte de la estructura de casi todos los sistemas operativos en que la relación directa con el usuario sea fundamental. Las otras dos partes del CP/M son el BDOS y el BIOS.





Un programa para facilitar las cosas

Un sistema operativo no es más que un programa que reside en la memoria temporal de un ordenador desde que lo activamos hasta que hacemos *RESET*. Su principal característica es que proporciona al programador un método estándar de acceder a los periféricos. Así, nuestros programas CP/M pueden pedirle al operativo que escriba una línea en la pantalla, sin tener que preocuparse de la organización concreta de la pantalla en la máquina donde corren. Los sistemas operativos gastan la mayor parte de su espacio en proporcionar un método eficiente de gestión de los archivos en disco, y por eso se llaman muchas veces DOS (Disk Operating System).

La segunda misión importante de un sistema operativo es asegurarle al usuario la posibilidad de introducir los comandos mínimos para cargar un programa desde el teclado, averiguar el contenido de un disco o realizar la estadística de uso del sistema. En muchos miniordenadores esta tarea se suele realizar por una parte del BASIC en ROM, pero no tiene una gran relación con éste. La parte del operativo que la realiza se llama CCP.

El BDOS (Basic Disk Operating

	FUNCIONES DEL CP/M		
		Parámetros	
	Registro	Entrada	Salida
0	RESET	Ning.	Ning.
1	Lect. Teclado (CON)	Ning.	Car ASCII en A
2	Esc. Teclado (CON)	ASCII en E	Ning.
3	Lect. Aux. (RDR)	Ning.	ASCII en A
4	Esc. Aux. (PUN)	ASCII en E	Ning.
5	Esc. Impr. (LST)	ASCII en E	Ning.
6 (1)	E/S teclado	ASCII en E SI E=FF leer	Ning. ASCII en E
9	Esc. Cadena	direc. en DE. Cadena acaba en $	Ning.
10 (1)	Leer línea	Direc. en DE (DE)=long.	Buffer lleno
11	Estado CON	Ning.	A=1 caracter A=0 No hay tecla pulsada
15 (1)	Abrir fichero	Direc. de FCB en DE	0080H+32*núm, en direct. FFH si no existe fich.
16	Cerrar fich.	FCB en DE	como 15
20 (2)	Leer registro	FCB en DE	a=0 correcto a=1 EDF. Otros errores
21 (2)	Esc. registro	FCB en DE	a=0 correcto otros valores errores
22 (2)	Crear fich.	FCB en DE	como en 15
23 (1)	Camb. nombre	FCB en DE	como en 15

(1) Añadidas o cambiadas en 2,0 y siguientes.
(2) Añadidas o cambiadas en 3,1.

System) y el BIOS (Basic Input Output System) son realmente el corazón del sistema operativo, y la clave del éxito de CP/M como un estándar. La explicación de su cometido resulta más fácil si utilizamos un mapa de memoria de nuestro sistema operativo (figura 1). El mapa en el CP/M Plus es ligeramente distinto, ya que éste maneja más de 64 K. Pero la idea es la misma.

Podemos observar que la parte baja de memoria (desde 0000H hasta la localidad 256 0100H) incluye una serie de variables y parámetros que nos permiten la comunicación con el sistema.

La localidad 0000 incluye el punto de entrada de RESET, y, en CP/M 2,2 y anteriores, vuelve a cargar el CCP y reinicializa los discos. Es el salto que se ejecuta cuando pulsamos ↑ C. En el Plus los discos no se inicializan de nuevo.

El punto de entrada para todas las funciones del operativo es la localidad 5, donde por regla general se encuentra un salto hacia el BDOS. El BDOS es la parte estándar del sistema operativo, y el BIOS es la parte que depende de la máquina, y que se debe escribir una vez para cada ordenador. Aunque el BIOS depende de la máquina, Digital Research proporciona un BIOS estándar, junto con instrucciones para su modificación «a la medida». Y esta no es una de las menores razones de su éxito.

El BDOS, como parte estándar del sistema, permite a los programadores utilizar una gran cantidad de puntos de entrada para acceder al sistema. Para los expertos en ensamblador indicaremos que, para conseguir acceso a las funciones del CP/M hay que llamar a la localidad ØØ5 con el número de función a utilizar en el registro C y parámetros opcionales en los registros A y DE. Los parámetros dependen de la llamada, y la tabla 2 muestra algunos de ellos.

La estructura del disco

Para poder controlar todos los parámetros del disco del sistema, el operativo tiene que especificar las características que debe reunir el directorio. En el CP/M se considera que los sectores tienen 128 bytes, aunque el sistema puede usar discos con sectores de 512 (los más corrientes) y hacer luego la conversión. Cada fichero ocupa un mínimo de una entrada de directorio, que ocupa 32 bytes. Los discos tienen unas pocas pistas reservadas para el directorio y, si son discos «de sistema», para las rutinas de carga del operativo. Si, por ejemplo, el directorio permite 64 archivos, ocupará los primeros 16 sectores tras los «del sistema».

Cada vez que se opera sobre un fichero o sobre su entrada de directorio, ésta se almacena temporalmente en memoria, actualizándose en disco al acabar el proceso. Por eso es importante que se cierren todos los ficheros antes de cambiar los discos.

Los nombres de archivo deben cumplir ciertas condiciones: el nombre pue-

de incluir hasta ocho caracteres, que serán convertidos a mayúsculas, y una extensión, separada por un punto, de tres letras. Algunos comandos permiten utilizar la «?» y el «*» para indicarle al operativo que en esa posición vale cualquier letra o grupo de letras.

La tabla de nombre de archivo tiene el siguiente formato (tabla 3). Los bits más significativos de los caracteres del nombre y tipo de fichero se utilizan para marcar información suplementaria, con algunos de ellos reservados para el sistema y otros de usuario. En ficheros de más de 16 K el fichero ocupa 2, 3..., extensiones de 16 K, y, por tanto, esas entradas de directorio. Aunque existen ligeras diferencias entre las versiones de CP/M en la forma de tratar los ficheros, la estructura no muestra grandes variaciones. No hay que olvidar que el CP/M se ha diseñado para una gran variedad de máquinas, y que se trata de un operativo que ha evolucionado durante más de diez años.

El CP/M Plus tiene bastantes diferencias con respecto a las versiones anteriores. Puesto que se ha construido para poder direccionar más de 64 K, el BIOS estándar no es más que una tabla de saltos que permite acceder, mediante un mecanismo de paginación, al programa ejecutivo, que está en las «otras» 64 K. Allí se encuentra también la memoria de pantalla, otra de las limitaciones importantes del CP/M cásico. En efecto, las pantallas de los ordenadores evolucionaron desde la simple pantalla de texto, que no ocupa apenas memoria, a la pantalla gráfica del Amstrad, que se «come» 16 K a cambio de los maravillosos gráficos que ofrece.

Esta memoria de pantalla, en el CP/M 2,2, está en la parte alta de la memoria, junto con el BIOS, lo que hace una memoria efectiva de 48 K. Con las 128 K y la paginación se dispone de un CP/M máximo, con un área de usuario de más de 61 K.

Los comandos residentes

Ya hemos dicho que, cuando aparece el «A>», el usuario puede introducir un comando. El intérprete de comandos mira a ver si la palabra introducida coincide con alguno de los comandos internos. En ese caso lo ejecuta. Si no se corresponde, busca en el disco un archivo del mismo nombre y de tipo COM (Comando), lo carga en memoria y lo ejecuta.



Digital Research, firma que ha hecho el CP/M, ofrece su gama de productos para Amstrad.

Los primeros comandos se llaman *residentes*, los segundos *transitorios.*

Los comandos residentes son:

ERA, que borra uno o más ficheros del disco.

DIR, que muestra información sobre los ficheros de usuario del disco.

DIRS, igual que DIR para los comandos del sistema.

REN, que cambia el nombre a un fichero.

SAVE, que permite crear un fichero con el contenido de la memoria.

TYPE, que escribe un fichero de texto en la pantalla.

USER, que permite dividir el disco en hasta 16 zonas de usuario.

Además, en sistemas de más de un disco, se puede cambiar el disco por defecto escribiendo la letra que lo identifica seguida por «:». Por ejemplo, el usuario de PCW 8256 puede mejorar mucho la velocidad de trabajo copiando los comandos transitorios que usa al disco de memoria «M:» y luego escribiendo tras el «A>» un «M:». El sistema contestará «M>» y, a partir de ese momento, buscará los comandos en el disco simulado en memoria. Otro comando que se puede utilizar es CONTROL P (abreviado P), pero sólo si dispone de impresora. Así hacemos que todo lo que aparece en pantalla vaya también a impresora. Una nueva pulsación de P desactiva la impresora.

Los comandos de disco

Los ficheros de disco cuya extensión sea COM se consideran como comandos del sistema, si tecleamos «Hola», el ordenador busca en el disco y acaba por contestar «HOLA?». Eso quiere decir que ha buscado el comando HOLA, que no es residente, en el disco, y que no existe un fichero HOLA, COM.

El número de ficheros transitorios depende de la instalación y de los discos de utilidad de que dispongamos. Por ejemplo, un programa de contabilidad se puede activar como CONTA, ya que estará en el fichero CONTA.COM.

Otro tipo de ficheros importantes para el programador son los ficheros de comandos, que ahorran introducir sucesivamente una serie de comandos del teclado. Estos ficheros tienen la extensión SUB, y se ejecutan mediante el comando SUBMIT. Los usuarios de LOGO3, en CP/M Plus, pueden ejecutarlo como LOGO3 (llamando al fiche-


Avda. de la Constitución, 260
Tels. 675 78 54 - 675 78 22
Torrejón de Ardoz
Madrid
PROTOMEC®
P.V.P. RECOMENDADO 1.700 pts.
Compatible con: SPECTRUM, ATARI, COMMODORE, SPECTRAVIDEO, AMSTRAD, CANON Y OTROS.
Spectrum, Atari, Commodore, Spectravídeo, Amstrad y Canon son marcas registradas por sus respectivos propietarios.
RECOMENDADO AMSTRAD
GARANTIZADO POR 3 MESES
PROTO®
Joystick

FCB (Bloque de Control de Archivos)

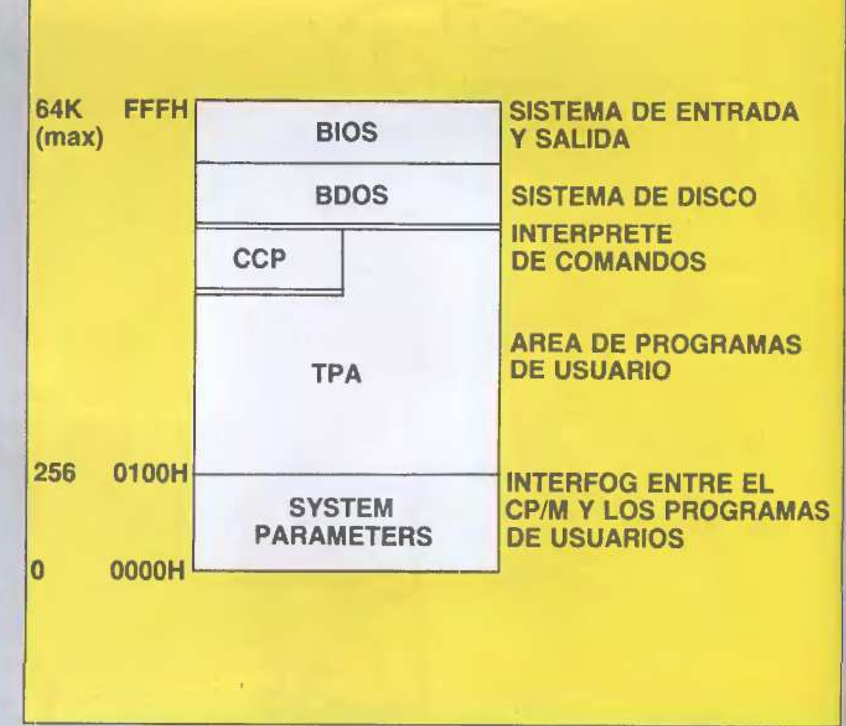
Byte	Nombre	Contenido
00	ET	Tipo de entrada. E5=libre o borrada On Num. de usuario
01-08	FN	Ocho caracteres de nombre, con espacios a la derecha.
09-0B	FD	Tipo de fichero. Los bits más significativos de cada byte sirven de indicadores de protección de escritura, etc.
OC	EX	Número de extensión. Cada 16K de fichero éste ocupa una nueva entrada en el directorio.
OD		No se usa CP/M 2,2
OE	S2	Es parte de la cuenta de extensiones.
OF	RC	Cuenta de registros. Una extensión puede tener hasta 128 registros de 128 bytes.
10-1F	DM	Mapa del Disco. Es una tabla con los bloques lógicos que contienen partes del fichero.
20	NR	Registro siguiente. El siguiente registro que se va a leer o escribir.
21-23		Se utilizan como puntero para el acceso directo.

ro COM) o bien como SUBMIT LOGO3, que ejecuta el fichero LOGO3. SUB, que a su vez contiene los comandos SETKEYS KEYS. DRL (para reasignar el teclado para LOGO) y LOGO3, que ejecuta el programa propiamente dicho.

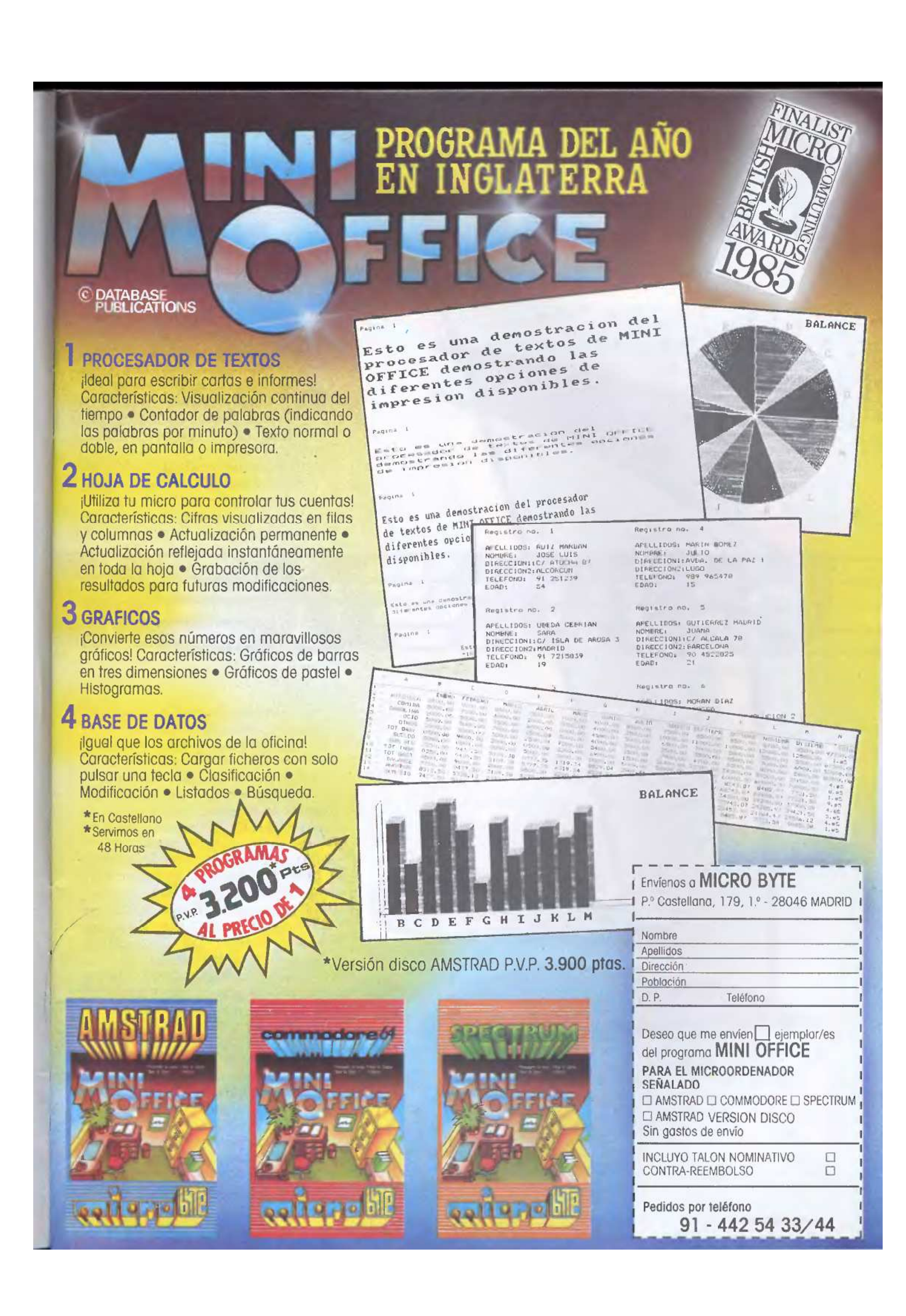
Muchas posibilidades, al precio de un poco de estudio

No hemos hecho más que subrayar unas pocas de las posibilidades del CP/M. Este operativo, pensado para los programadores, incluye además un monitor de código máquina (DDT o SID), un editor (ED), rutinas para volcar un fichero en hexadecimal (DUMP), etcétera.

Una exploración exhaustiva de todas las posibilidades agotaría el espacio de la revista, y más aún si dispusiéramos de él. Existen buenos libros sobre el tema, que comentaremos en un próximo número, y seguiremos tratando este segundo operativo, que puede convertirse en el alma de nuestro ordenador profesional.







MINI OFFICE
PROGRAMA DEL AÑO EN INGLATERRA
FINALIST MICRO COMPUTING AWARDS BRITISH 1985
© DATABASE PUBLICATIONS
1 PROCESADOR DE TEXTOS
¡Ideal para escribir cartas e informes! Características: Visualización continua del tiempo • Contador de palabras (indicando las palabras por minuto) • Texto normal o doble, en pantalla o impresora.
2 HOJA DE CALCULO
¡Utiliza tu micro para controlar tus cuentas! Características: Cifras visualizadas en filas y columnas • Actualización permanente • Actualización reflejada instantáneamente en toda la hoja • Grabación de los resultados para futuras modificaciones.
3 GRAFICOS
¡Convierte esos números en maravillosos gráficos! Características: Gráficos de barras en tres dimensiones • Gráficos de pastel • Histogramas.
4 BASE DE DATOS
¡Igual que los archivos de la oficina! Características: Cargar ficheros con solo pulsar una tecla • Clasificación • Modificación • Listados • Búsqueda.
*En Castellano
*Servimos en 48 Horas
4 PROGRAMAS P.V.P. 3.200* Pts AL PRECIO DE 1
Esto es una demostracion del procesador de textos de MINI OFFICE demostrando las diferentes opciones de impresion disponibles.
BALANCE
Registro no. 1
APELLIDOS: RUIZ MANDAN
NOMBRE: JOSE LUIS
DIRECCION1:C/ ATOCHA 87
DIRECCION2:ALCORCON
TELEFONO: 91 251239
EDAD: 54
Registro no. 2
APELLIDOS: UBEDA CEBRIAN
NOMBRE: SARA
DIRECCION1:C/ ISLA DE AROSA 3
DIRECCION2:MADRID
TELEFONO: 91 7215839
EDAD: 19
Registro no. 4
APELLIDOS: MARIN GOMEZ
NOMBRE: JULIO
DIRECCION1:AVDA. DE LA PAZ 1
DIRECCION2:LUGO
TELEFONO: 989 965478
EDAD: 15
Registro no. 5
APELLIDOS: GUTIERREZ MADRID
NOMBRE: JUANA
DIRECCION1:C/ ALCALA 78
DIRECCION2:BARCELONA
TELEFONO: 90 4522025
EDAD: 21
Registro no. 6
APELLIDOS: MORAN DIAZ
BALANCE
B C D E F G H I J K L M
*Versión disco AMSTRAD P.V.P. 3.900 ptas.
Envíenos a MICRO BYTE
P.º Castellana, 179, 1.º - 28046 MADRID
Nombre
Apellidos
Dirección
Población
D. P.
Teléfono
Deseo que me envíen ☐ ejemplar/es del programa MINI OFFICE
PARA EL MICROORDENADOR SEÑALADO
☐ AMSTRAD ☐ COMMODORE ☐ SPECTRUM
☐ AMSTRAD VERSION DISCO
Sin gastos de envío
INCLUYO TALON NOMINATIVO ☐
CONTRA-REEMBOLSO ☐
Pedidos por teléfono
91 - 442 54 33/44
AMSTRAD
commodore 64
SPECTRUM
MINI OFFICE
microbyte

FIRMWARE

GESTOR DE SO

El Amstrad incorpora, como muchos de vosotros ya sabréis, un procesador especial para controlar el sonido. Se trata del AY-3-8912, hermano del que llevan los MSX, y un instrumento muy potente. Pero este circuito no servirá de nada sin un programa que le saque rendimiento. El programa se llama gestor de sonido (sound manager), y es el tema de nuestro capítulo de hoy.

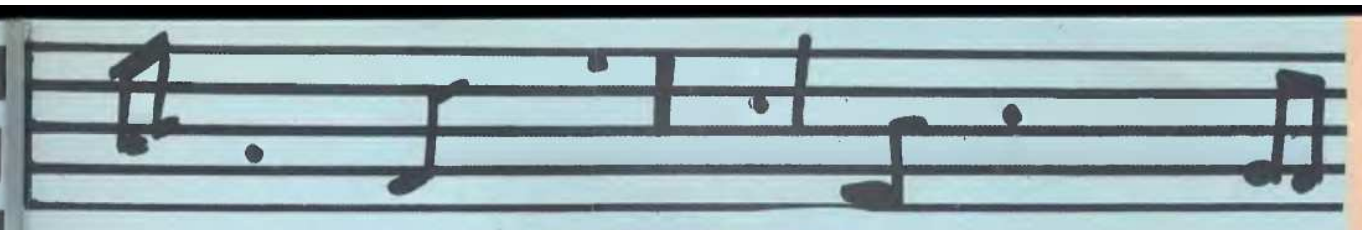
La salida de sonido de los CPC es una clavija estéreo, aunque su sonido es habitualmente mono por llevar un solo altavoz en el interior de la máquina. Se trata de un pequeño altavoz, que le hace poca justicia a las posibilidades sonoras del ordenador.

La pastilla que incorpora el Amstrad tiene realmente tres canales. Suponiéndolos numerados como canal 1, canal 2 y canal 3, la salida al jack estéreo se da como canal 1 más canal 2 por la salida izquierda, y el canal 2 y canal 3 por la salida derecha.

No obstante las facilidades del Amstrad no terminan aquí siendo posible gestionar esta pastilla, accediendo directamente a ella mediante comandos firmware o si se prefiere se puede acceder usando el software disponible por Locomotive.

Vamos a estudiar superficialmente la funcionalidad de los parámetros a introducir a nuestro Amstrad.

Como es bien sabido el sonido no es más que una onda, por lo que resulta obvio que su comportamiento ha de seguir las mismas reglas de cualquier movimiento armónico. Empecemos por ver la amplitud de la onda. La amplitud nos define realmente el volumen del sonido, aunque combinando los diferentes parámetros podemos conseguir efectos de más o menos volumen. Si introducimos pues el valor



cero a este parámetro está claro que no vamos a oir nada. Los valores disponibles para este parámetro oscilan entre 0 y 15. La frecuencia es el número de veces que se repite una nota en la unidad específica de tiempo. El período viene generado por la siguiente fórmula:

$$PERIODO = \frac{125.000}{FRECUENCIA}$$

Haciendo una pequeña referencia a la estructura de una nota musical, se pueden diferenciar tres fases en la ejecución de la misma, esto es:

Fase 1: fase de crecida en la que la nota va «creciendo»



Fase 2: fase de sostenida o fase en la cual la nota se ejecuta y mantiene.

Fase 3: fase de decaimiento en la cual una vez terminada la ejecución de la nota ésta todavía tarda un tiempo en desaparecer del todo.

Con el fin de igualar al máximo las posibilidades de un instrumento de música, nos está permitido controlar la duración de las tres fases expuestas anteriormente.

Potente, pero complejo

La complejidad de la gestión del sonido del chip que incorporan los Amstrand hace que el operativo facilite su programación mediante interrupciones. Aunque para hacer sonar la máquina hasta proporcionarle una frecuencia y una duración, un control más completo del sonido nos obliga a poner en juego muchos más parámetros.

Peró el estudio de todos los parámetros nos llevaría un artículo de mayor longitud que éste por lo que dejamos el resto de los parámetros y un estudio mucho más profundo para una

Tabla

Llamada	Función	Notas
BCA7	Resetea el Sound Manager	
BCAD	Testea el estado de un canal	A: contiene un canal según: 1000000 Canal A 01000000 Canal B 00100000 Canal C A: contendrá en salida el estado del canal según BITS significan: 0 1 2.n.° de notas libres en el canal 3. espera de sincronism Con canal A 4. espera de sincronism Con canal B 5. espera de sincronism Con canal C 6. el canal esta «sostenido» 7. el canal está activo
BCB3	Libera un canal «sostenido»	A contiene el canal según çuadro anterior
BCB6	Para todos los sonidos de forma inmediata	Retiene todos los canales pero sin perder la información
BCB9	Consigue la dirección de la envolvente de amplitud	A contiene la envolvente por tanto ha de ser un número de 1 a 15. HL contendrá la dirección de la envolvente de amplitud. BC contendrá la longitud de la envolvente.
BCC5	Consigue la dirección de la envolvente de tono	Según el cuadro anterior.
BC65	Inicializa el cassete manager	
BC6B	Activa o desactiva los mensajes del cassette	A si cero son activados. A si uno son desactivados
BC6E	Activa el motor	
BC71	Desactiva el motor	
BC77	Abre un fichero como input	B: contiene la longitud del fichero HL: contiene la dirección del fichero DE: contiene la dirección del buffer de salida
BC7A	Cierra un fichero	
BC80	Lee un caracter de un fichero de entrada	A: contiene el caracter leído
BC89	Testea el fin de fichero	Si es final Carry True
BC8C	Abre un fichero de salida	Ver cuadro de BC77
BC95	Escribe un caracter	A: contiene el caracter
BCA1	Lee un registro del cassette	HL: contiene la dirección donde dejar los datos. DE: contiene la longitud de los datos. A: contiene un caracter de sincronismo según: Pt2C Registro cabecera Pt16 Registro de datos
BC9E	Graba un registro	Igual que en cuadro anterior
BCA4	Verifica un registro	Igual que en cuadro anterior.



Un buen lector de cabeceras es la clave de cualquier sistema de recuperación de información en cassette.

EJEMPLO DE GESTION DE CABECERAS

```
                     10 ;CONVERSION DE UN BLOQUE CUALQUIERA DE UN PROGRAMA A
                     20 ;BLOQUE DE CARGA DE DATOS O PRIMER BLOQUE DE CARGA
29BE  CD18BB         30 INICIO: CALL #BB18             ; ESPERA UNA TECLA PARA L
EER REGISTRO
                     40 ;DE CABECERA
29C1  218813         50         LD   HL,5000           ;POSICION DE CARGA DE DAT
OS EN MEMORIA
29C4  110001         60         LD   DE,256            ;LONGITUD DEL REGISTRO CA
BECERA
29C7  3E2C           70         LD   A,#2C             ; CARACTER DE SINCRONISMO
 PARA CABECERA
29C9  CDA1BC         80         CALL #BCA1             ; LECTURA DE UN REGISTRO
                     90 ;----------------------------------------------------------
---
                    100 ;TENEMOS EL REGISTRO CABECERA EN LA POSICION 50000
                    110 ;----------------------------------------------------------
---
29CC  3E7F          120         LD   A,%1111111
29CE  329F13        130         LD   (5023),A
                    140 ;----------------------------------------------------------
---
                    150 ;UN VALOR DIFERENTE DE 0 LO CONVIERTE EN BLOQUE DE CARGA
                    160 ;----------------------------------------------------------
---
                    170 ;DISPONEMOS DEL REGISTRO MODIFICADO
                    180 ;----------------------------------------------------------
---
29D1  CD18BB        190         CALL #BB18
29D4  218813        200         LD   HL,5000
29D7  110001        210         LD   DE,256
29DA  3E2C          220         LD   A,#2C
29DC  CD9EBC        230         CALL #BC9E             ; GABACION DE UN REGISTRO
.
29DF  C9            240         RET
```

PERSONAL PRINTER

:RITEMAN F+:

||| NEAR LETTER QUALITY PRINTER

COMPATIBILITY WITH MOST PERSONAL COMPUTERS



:RITEMAN F+:

ESPECIFICACIONES

1. **Método de impresión**
Serie de impactos por matriz de 9×9

2. **Cabezal impresor**
9 agujas (reemplazable) con expectativa de vida de 50 millones de caracteres a 100 % de uso contínuo.

3. **Velocidad de impresión**
Impresión a 105 cps uni o bi-direccional, búsqueda lógica optimizada, capacidad de impresión 45 lpm a 10 cpi., 200 ms de salto de línea con espaciado de 1/6".

4. **Espaciado interlínea**
1/6", 1/8", 7/72", programable N/72", N/216".

5. **Alimentación de papel**
a) Tractor ajustable entre 4" y 10".
b) Alimentación a fricción para hojas sueltas entre 4" y 9,5".

6. **Corte de papel**
Se puede imprimir desde principio de página hasta una pulgada de fin de papel.

7. **Selección de caracteres**
Todos los 96 caracteres ASCII, 96 itálicos, 96 NLQ, todos ellos con correcta impresión de los trazos descendentes, y 9 set de diferentes idiomas seleccionables.
Además con 2K buffer, 32 caracteres definibles por usuario y con 8K buffer, 256 caracteres definibles por usuario.

8. **Gráficos**
Densidad normal 480 puntos por línea, doble densidad 960, cuadruple densidad 1920, CRY gráficos 640, plotter gráfico 576, CRT gráficos 720 puntos por línea.

9. **Modos de impresión**
Standard, doble impresión, enfatizado, itálica, supra y subíndices, mitad altura y NLQ.

10. **Tamaños de impresión**
a) Normal (10 cpi) 2,1 mm (a) × 2,55 mm (h)
b) Normal expandido (5 cpi)
c) Elite (12 cpi)
d) Elite expandido (6 cpi)
e) Comprimido (17 cpi)
f) Comprimido-expandido (8,5 cpi)
g) Espaciado proporcional
h) Proporcional expandido.

11. **Test escritura**
Manteniendo apretado LF (salto de línea) cuando se acciona el interruptor de puesta en marcha ON.

12. **Subrayado contínuo.**

13. **Tabulador**
Existen 32 posiciones de tabulación por línea.

14. **Longitud de formato**
El standard de 66 líneas, con espaciado de líneas y longitud del formato programable.
VFU eléctrico.

15. **Tamaño buffer**
1 línea standard.

16. **Buffer RAM**
2 K byte (standard)
8 K byte (opcional).

17. **Tope posicionador de papel contínuo, desplazable parte anterior o posterior.**

18. **Interface**
Standard paralelo centronics de 8 bits.
Serie RS232 C opcional.

19. **Grosor papel**
Máximo 2 copias, 40 g/m² peso contínuo, sensor papel.

20. **Operación con interruptores**
De fácil acceso.

21. **Caracteres por línea**
Normal: 80
Normal expandido: 40
Comprimido: 132
Comprimido expandido: 66
Elite: 96
Elite expandido: 48
Proporcional: n

22. **Cinta impresora**
Autorretintada, mediante tampón interno, cartucho compacto de cinta negra fácilmente reemplazable y bajo coste.

23. **Tamaño**
115 (h) × 275 (f) × 405 (a).

24. **Tensión alimentación**
220 v/ 50 Hz.

25. **Entrada de papel**
Frontal-horizontal.

26. **Retroceso**
Por software.

27. **Bajo nivel de ruido**
2 velocidades:
Velocidad lenta de 52 cps.
Velocidad rápida de 105 cps.

28. **Elevadores inferiores standard**
(para ubicación del papel contiuo debajo de la impresora).

29. **Motor: paso a paso**

30. **Indicadores:**
• Conexión general
• Fin de papel
• On line

31. **Interruptores**
ON/OFF, On line (SEL), salto de página (FF), salto de línea (LF).

32. **Compatible con la mayor parte de software existente.**

TODAS LAS ESPECIFICACIONES PUEDEN SER CAMBIADAS SIN PREVIO AVISO.

DATAMON

DATAMON, S. A.
REPRESENTACION EN ESPAÑA DE:
:RITEMAN:
-IMPRESORAS PROFESIONALES-
PROVENZA, 385-387, 6.°, 1.°
Tel. (93) 207 27 04 08025 BARCELONA



próxima edición, reservando el resto del espacio para dar un pequeño repaso a las indirecciones del firmware.

Cassette manager

El cassette manager como su nombre indica es el gestor y responsable directo del funcionamiento del cassette, tanto en lo referente al modelo CPC-464, como a cualquier cassette conectado a cualquiera de los demás modelos.

Este sistema gestiona tanto la estructura de los datos en cinta como su fiabilidad de uso. Como se explicó en su momento Amstrad graba los datos por bloques, que constan de dos registros: registro de cabecera y registro de datos. El primero es el encargado de avisar al Amstrad lo que se va a leer, de qué tipo es el fichero, si está protegido, etc. El funcionamiento de este sistema es tan perfecto que si se varía en lo más mínimo, los efectos pueden ser desastrosos. Por ejemplo un tipo de protección incluido en un juego comercial, ha consistido en anular el registro de cabecera del bloque y aumentar la longitud del registro de datos, con los resultados de que con cintas originales del mismo producto el sistema de carga falla en un 60 por ciento de las veces con lo cual han conseguido anular una de las mejores características del Amstrad; su magnífica gestión de las cintas que en un 99 por 100 de las veces funcionaba sin ningún tipo de problema.

Retomando el hilo de nuestro artículo, en esta entrega vamos a intentar dar una serie de datos complementarios al artículo anteriormente publicado sobre el cassette.

Hasta dos ficheros simultáneos

El Amstrad es capaz de gestionar dos ficheros de cinta abiertos simultáneamente; uno de ellos como entrada de datos y otro como salida, lo cual permite hacer un sistema de gestión bastante completo teniendo en cuenta las limitaciones propias de una cinta.

Los nombres de los ficheros en cinta permiten hasta 16 caracteres con signos entre caracteres, pero lo recomendable es dar nombres de máximo ocho caracteres para así poder pasar nuestros programas a disco sin excesiva complicación. La estructura de los registros, como todos recordaréis fue comentada en el número 2 de nuestra revista, por lo que para el siguiente ejemplo es mejor que nos remitamos a ella.



La mayor colección de programas para AMSTRAD



3D STUNT RIDER.



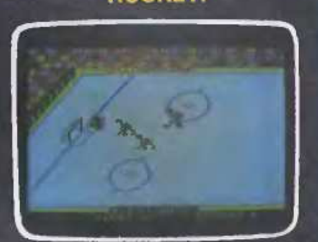
Participa en uno de los más excitantes deportes con riesgo: el motorismo acrobático. ¡Intenta batir el record mundial de salto sobre autobuses!
P.V.P. CASSETTE: 1.600 pts.

BRAXX BLUFF.



Lánzate a explorar mundos desconocidos con los Centuriones. En cada lugar que aterrices te aguardan peligros y desastres inesperados.
P.V.P. CASSETTE: 1.6000 pts.

HOCKEY.



Pon a prueba tus reflejos, rapidez y decisión compitiendo en un emocionante partido de hockey sobre hielo, contra un amigo o contra tu AMSTRAD.
P.V.P. CASSETTE: 1.600 pts. DISCO: 2.500 pts.

ASALTO.



Si te gusta la emoción de la guerra, desarrolla toda una batalla aeronaval. Tus enemigos no te darán cuartel y no olvides lo más importante: ¡atacar!
P.V.P. CASSETTE: 1.600 pts.

AIRWOLF.



Cinco científicos están retenidos en las profundidades de una base. ¿Podrás rescatarlos pilotando el inigualable y sofisticado helicóptero?. ¡Destruye las cajas de control de la defensa!
P.V.P. CASSETTE: 1.600 pts. DISCO: 2.500 pts.

DRAGONS.



Apasiónate en una inquietante aventura aniquilando dragones. Camina, trepa y salta para salvar tu vida de los dragones rojos.
P.V.P. CASSETTE: 1.600 pts.

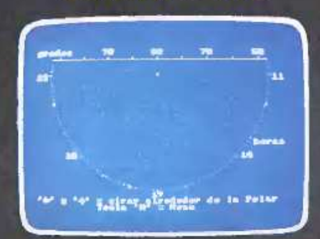
GRAND PRIX RALLY II.



Participa en el mundo de los Rallies con circuitos en tres dimensiones. Atraviesa hielo, lluvia, desiertos, niebla y evita los choques con tus competidores. Sorpréndete creando tus propios recorridos.
P.V.P. CASSETTE: 1.600 pts. DISCO: 2.500 pts.

EDUCA-
TIVOS

CONSTELACIONES.



Observa las más importantes estrellas del universo en su posición real (grados y horas). Descubre sus nombres y aprende a identificarlas y conocerlas.
P.V.P. CASSETTE: 1.600 pts.

ESTIMATOR RACER.



Desarrolla tu capacidad de cálculo numérico. Conduce tu coche por el carril, en una carrera contra reloj, con la respuesta matemática más aproximada. 4 niveles de dificultad.
P.V.P. DISCO: 2.500 pts.

EL CUERPO HUMANO.
EL ESQUELETO.



¿Cuántos nombres de huesos de nuestro esqueleto eres capaz de recordar? Aprende y diviértete con este excelente programa educativo.
P.V.P. CASSETTE: 1.600 pts.



ESPAÑA

Avda. del Mediterráneo, 9. Tels. 433 45 48 - 433 48 76. 28007 MADRID   Delegación Cataluña: Tarragona, 110 - Tel. 325 10 58. 08015 BARCELONA



- Envíos a domicilio a toda España
- Sin gastos de envío
- Servicio rápido

JUEGOS

Ref.	Titulo	P.V.P.
AMC-100	HOME RUNNER	1.600.-
AMC-101	GRAND PRIX	1.600.-
AMC-102	HARRIER ATTACK	1.600.-
AMC-103	SPANNERMAN	1.600.-
AMC-104	AGUILAS DEL ESPACIO	1.600.-
AMC-106	LA PULGA	1.600.-
AMC-107	FRED	1.600.-
AMC-108	MASTERCHESS	1.600.-
AMC-109	LASERWARP	1.600.-
AMC-110	HAUNTED EDGES	1.600.-
AMC-111	CODENAME MAT	1.600.-
AMC-112	AMSGOLF	1.600.-
AMC-113	HUNCHBACK (QUASIMODO)	1.900.-
AMC-114	HUNTER KILLER	1.600.-
AMC-115	ROLAND EN EL TIEMPO	1.600.-
AMC-116	ATOM SMASHER	1.600.-
AMC-117	ELECTRO FREDDY	1.600.-
AMC-118	ADMIRAL GRAPH SPEE	1.600.-
AMC-119	COMANDO ESTELAR	1.600.-
AMC-120	MANIC MINER	1.900.-
AMC-121	ASTRO ATTACK	1.600.-
AMC-122	QUACK	1.600.-
AMC-123	BILLAR (SNOOKER)	1.900.-
AMC-124	ROLAND EN EL TREN	1.600.-
AMC-125	LA PLAGA GALACTICA	1.600.-
AMC-126	ROLAND EN EL INFIERNO	1.600.-
AMC-127	SIMULADOR DE VUELO 377	2.000.-
AMC-128	ROLAND AHOY	1.600.-
AMC-129	EL PUENTE	1.600.-
AMC-130	MR. WONG	1.600.-
AMC-132	MUTANT MONTY	1.600.-
AMC-133	TENIS	1.600.-
AMC-134	LABERINTO DEL SULTAN	1.600.-
AMC-135	PUNCHY	1.600.-
AMC-136	ALIEN BREAK	1.600.-
AMC-137	FRUIT MACHINE	1.600.-
AMC-138	OH MUMMY	1.600.-
AMC-139	SPLAT	1.600.-
AMC-140	CRAZY GOLF	1.600.-
AMC-141	CUBIT	1.600.-
AMC-142	3D INVASORES	1.600.-
AMC-143	EL PREMIO	1.600.-
AMC-144	JET BOOT JACK	1.600.-
AMC-145	ROLAND EN EL ESPACIO	1.600.-
AMC-146	PYJAMARAMA	1.600.-
AMC-147	BOY SCOUT	1.600.-
AMC-148	FANTASTIC VOYAGE	1.600.-
AMC-149	THE KEY FACTOR	1.600.-
AMC-150	JAMMIN	1.600.-
AMC-151	SUPER PIPELINE	1.600.-
AMC-152	TRAFFIC	1.600.-

Si no encuentra el programa que está buscando, el periférico que necesita o el libro que le apetece...

Tenemos todo para su AMSTRAD

AMC-153	CATASTROPHES	1.600.-
AMC-154	SORCERY	1.600.-
AMC-155	ROLAND Y LOS CUBOS	1.600.-
AMC-156	FRANK'STEIN	1.600.-
AMC-157	GATE CRASHER	1.600.-
AMC-158	EL JUEGO DE LOS NUMEROS	1.600.-
AMC-159	HOCKEY	1.600.-
AMC-160	AIR WOLF	1.600.-
AMC-161	GRAN PRIX RALLY II	1.600.-
AMC-162	SUBTERRANEAN STRYKER	1.600.-
AMC-163	ALIEN	1.600.-
AMC-164	3D STUNT RIDER	1.600.-
AMC-165	DRAGONS	1.600.-
AMC-166	BRAXX BLUFF	1.600.-
AMC-167	HOUSE OF USHER	1.600.-
AMC-168	DEFEND OR DIE	1.600.-
AMC-169	AJEDREZ TRIDIMENSIONAL (Serie Oro)	2.300.-
AMC-170	3D GRAND PRIX (Serie Oro)	2.300.-
AMC-171	3D BOXING (Serie Oro)	2.300.-
AMC-172	SUPER TRIPPER (Serie Oro)	2.300.-

EDUCATIVOS

Ref.	Titulo	P.V.P.
AMC-300	ANIMAL, VEGETAL, MINERAL	1.600.-
AMC-301	PLANETARIUM/TIERRA	1.600.-
AMC-302	RIOS/CAPITALES	1.600.-
AMC-303	PERSONAJES/CRONO	1.600.-

PROFESIONALES

Ref.	Titulo	P.V.P.
AMC-500	Contabilidad Personal	3.000.-
AMC-501	AMSCALC (Hoja Cálculo)	4.300.-
AMC-502	AMSWORD (Proceso Textos Simple)	2.000.-
AMC-503	AMSWORD II. Proceso de Texto	4.900.-
AMC-504	AMSBASE. Base de Datos	2.000.-
AMC-505	STOCK-AID	1.900.-
AMC-506	GENERADOR DE FACTURAS, FICHAS Y DOCUMENTOS (Invostat)	1.900.-
AMC-507	BASE DE DATOS Y ETIQUETAS	1.900.-
AMC-508	TRATAMIENTO DE FICHEROS (MasterFile)	4.300.-
AMC-509	EL EMPRESARIO Instruc. para dirigir su negocio	4.300.-
AMC-510	HOJA DE CALCULO (MasterCalc)	5.000.-

UTILIDADES Y LENGUAJES

Ref.	Titulo	P.V.P.
AMC-700	DEVPAC. ENSAMBL/DESENSAMBLADOR	4.300.-
AMC-701	HISOFT PASCAL	4.900.-
AMC-702	Diseñador de Pantallas	4.300.-
AMC-703	AMSDRAW Gráficos y dibujos	2.000.-

AMC-704	GUIA DE PASCAL (libro)	7.500.-
AMC-705	FORTH	5.000.-
AMC-706	AMSTEST	1.600.-

JUEGOS

Ref.	Titulo	P.V.P.
AMD-100	FRED/LA PLAGA GALACTICA	4.900.-
AMD-101	LA PULGA/AMSDRAW	3.000.-
AMD-102	ROLAND EN EL ESPACIO	2.500.-
AMD-103	SORCERY	2.500.-
AMD-104	PYJAMARAMA	2.500.-
AMD-105	EL JUEGO DE LOS NUMEROS	2.500.-
AMD-106	HOCKEY	2.500.-
AMD-107	AIRWOLF	2.500.-
AMD-108	GRAN PRIX RALLY II	2.500.-
AMD-109	SORCERY PLUS (Serie Oro)	3.000.-
AMD-110	AJEDREZ TRIDIMENSIONAL (Serie Oro)	3.000.-
AMD-111	3D GRAND PRIX (Serie Oro)	3.000.-
AMD-112	3D BOXING (Serie Oro)	3.000.-
AMD-113	BEACH HEAD (Serie Oro)	3.000.-
AMD-114	SUPER TRIPPER (Serie Oro)	3.000.-

EDUCATIVOS

Ref.	Titulo	P.V.P.
AMD-300	MIRANDO LAS ESTRELLAS	4.300.-
AMD-301	PLANETARIUM/CRONO/TIERRA	2.500.-
AMD-302	CAPITALES/RIOS/PERSONAJES	2.500.-

PROFESIONALES

Ref.	Titulo	P.V.P.
AMD-500	PLANIFICACION DE PROYECTOS (Project Planner)	4.900.-
AMD-501	TOMANDO DECISIONES (Decision Maker)	4.900.-
AMD-502	BASE DE DATOS CP/M (Micropen)	10.500.-
AMD-503	TRATAMIENTO DE TEXTOS CP/M (MicroScript)	10.500.-
AMD-504	TRATAMIENTO DE TEXTOS (Amsword)	5.700.-
AMD-505	CONTABILIDAD GENERAL	12.700.-
AMD-506	CONTROL DE STOCK	10.500.-
AMD-507	CONTROL DE VENCIMIENTOS	7.500.-
AMD-508	HOJA DE CALCULO CP/M (MicroSpread)	10.500.-
AMD-509	EL EMPRESARIO (Instruc. para dirigir su negocio)	5.700.-
AMD-510	MASTERCALC	5.700.-
AMD-511	TRATAMIENTO DE TEXTOS PROFESIONAL (Amsword II)	5.700.-

UTILIDADES Y LENGUAJES

Ref.	Titulo	P.V.P.
AMD-700	HISOFT PASCAL	5.700.-
AMD-701	ENSAMBLADOR/DESENSAMBLADOR (DevPac)	5.700.-
AMD-702	DISEÑADOR DE PANTALLAS (Screen Designer)	4.900.-
AMD-703	GUIA DE PASCAL	5.700.-
AMD-704	FORTH	5.700.-

COMPLEMENTOS

Ref.	Titulo	P.V.P.
AMD-900	DISCO VIRGEN 3"	1.050

LIBROS EN CASTELLANO

Ref.	Titulo	P.V.P.
AML-100	Curso Autod de Basic I	2.500.-
AML-101	Programando con Amstrad	2.100.-
AML-102	Juegos sensacionales para Amstrad	1.700.-
AML-103	40 Juegos Educativos para Amstrad	1.700.-
AML-104	Guia de Referencia Basic para el Programador	3.000.-
AML-105	Música y Sonidos con el Amstrad	1.000.-
AML-106	Hacia la Inteligencia Artificial con Amstrad	1.300.-
AML-107	Código máquina para principiantes con Amstrad	1.900.-
AML-108	Curso autodidáctico Basic II	2.500.-
AML-109	Programando Basic con AMSTRAD	1.900.-
AML-110	Técnicas de Programación de Gráficos en el Amstrad	1.700.-
AML-500	Manual de Firmware (en inglés)	3.600.-

PERIFERICOS Y ACCESORIOS

Ref.	Titulo	P.V.P.
AMP-100	Unidad de Diskette con controlador	45.500.-
AMP-101	Unidad de Diskette sin controlador	39.500.-
AMP-102	Modulador TV para 464	9.000.-
AMI-103	Impresora	45.700.-
AMP-104	Cable Centronics	3.675.-
AMI-109	Impresora Printer 130	54.900.-
AMP-105	Cable 2.ª unidad disco	3.450.-
AMP-106	Cable Cassette 664 (audio)	1.150.-
AMP-107	Modulador TV para 664 y 6128	9.450.-
AMP-108	Sintetizador de Voz SSA-1	9.000.-
AMP-109	Interface Serie RS 232 C	11.750.-
SVP-116	JOYSTICK QUICKSHOT I	1.600.-
SPV-117	JOYSTICK QUICKSHOT II	2.500.-
SVP-133	JOYSTICK QUICKSHOT VII	2.200.-

BOLETIN de PEDIDO

Ref.	TITULO	Cant.	P. V. P.	TOTAL
	PRECIO TOTAL PESETAS			

Nombre

Apellidos

Dirección

Población

D.P. Teléfono

Incluyo talón Bancario a nombre de "CHIPS & TIPS" ☐

Contra Reembolso ☐

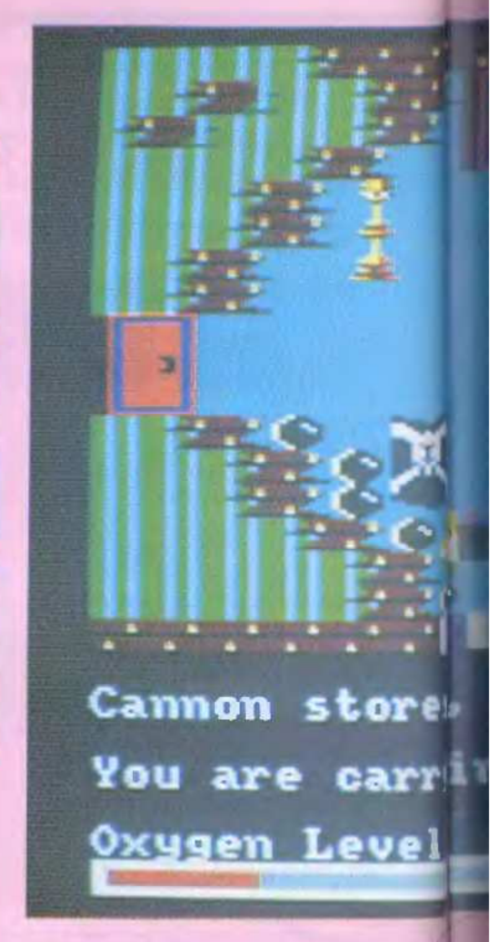
ENVIEN ESTE BOLETIN A:



"CHIPS & TIPS", Paseo de la Castellana, 126 2046 MADRID
Pedidos por teléfono (91) 411 64 63

"SOFTMAIL" Es un servicio de "CHIPS & TIPS"

The Devil's Crown



Programa: The Devil's Crown
Tipo: juego
Distribuidor: Serna
Formato: disco y cassette
Precio:

De manos de Probe Soft, nueva empresa en juegos para Amstrad, llega un gran programa para quien disfrute con las aventuras animadas y de acción. Del mismo tipo que el Sorcery, y aunque menos cuidado en el terreno gráfico, ofrece un argumento mucho más original. Montados en un minisubmarino y explorando cada camarote de un pecio pirata, deberemos encontrar la legendaria Corona del Diablo, una valiosísima corona de oro que descansa desde hace muchos años en el fondo de los mares.

En las entrañas de un barco hundido

El mapa incluye unas 40 pantallas distintas, por las que podremos movernos siempre que tengamos suficiente oxígeno. En ellas encontraremos diferentes objetos, todos los cuales nos serán de suma utilidad si los utilizamos en el momento adecuado. Para cogerlos hay que pulsar el botón de disparo cuando pasamos sobre ellos; dadas las limitaciones de potencia de nuestro pequeño vehículo sólo podremos llevar uno cada vez. En la parte baja de la pantalla se nos indica (en inglés) cuál es el camarote donde estamos, en qué cubierta se encuentra y qué objetos transportamos en ese momento. Un objeto importante es la linterna, sin la que no veremos nada en algunos de los camarotes.

A lo largo del juego nos encontraremos con algunos enemigos. Los más curiosos son unos peces (no de IBM), que pueden llegar a atacarnos si no les disparamos, pero que, en cuanto lo hacemos, huyen despavoridos. También están los fantasmas de la tripulación pirata que llevaba el barco, que pueden resultar más o menos peligrosos según el objeto que transportemos. Por ejemplo, un escudo los detendrá, una daga los destruirá, y si sobra llevamos una campana moriremos. Los fantasmas pueden influir en nuestro nivel de oxígeno, aumentándolo o disminuyéndolo según les dé.

En la parte inferior de la pantalla también hay un contador que marca la cantidad de oxígeno de que disponemos. Hay que estar atento



y buscar alguna escafranda autónoma (léase bombona de oxígeno) en cuanto lleguemos a la zona roja, pues si se termina el aire (evidentemente) nos ahogaremos. Otro contador señala las balas que nos quedan, cuando este número sea bajo podremos aumentarlo (hasta un máximo de 50) cogiendo alguna pistola de las que se encuentran por el barco.

Dos bombas y un destino

Llegar hasta la corona no es nada fácil, tendremos que seguir una serie de fases antes de que la podamos ver. Primero hay que coger las tres campanas que se encuentran por el barco y llevarlas a una habitación donde hay otras tres parpadeantes. Después hay que recoger los zafiros. Esto no es tan fácil como pueda parecer, pues es necesario llevar un determinado objeto, que podremos cambiar por uno de ellos. Para finalizar hay que dejar la llave adecuada sobre el candado adecuado para que la Corona del Diablo aparezca.

Todo esto ha de hacerse muy rápidamente, pues las botellas de oxígeno que hay repartidas por toda la nave se irán acabando, y con ellas nuestra existencia. Los simpáticos, aunque peligrosos, peces morirán definitivamente cuando les disparemos, pero los fantasmas sólo desaparecen temporalmente cuando los destruimos, por lo que en cuanto volvemos a entrar en el camarote donde se encontraban aparecen de nuevo.

Hay que tener mucho cuidado con las banderas piratas que podamos encontrar, pues suelen resultar bastante peligrosas si intentamos cogerlas. Conviene, por tanto, ignorarlas y pasar de largo.

Gráficos y sonido

El movimiento de los sprites es rápido pero suave, con un efecto bastante bueno. El único defecto es que la rutina de impresión no está controlada por interrupciones, y además borra y vuelve a dibujar el sprite cada vez aunque éste no se haya movido. El resultado de todo esto es un parpadeo constante que puede resultar algo molesto, aunque no tanto como para hacernos abandonar un juego tan atractivo.

Se puede disparar en la dirección en que se avanza incluyendo las diagonales y excluyendo arriba y abajo. Las ráfagas son rápidas y nutridas, por lo que resulta divertido utilizarlas. Los peces tienen unas reacciones bastante curiosas, no se mueven al azar pero tampoco son los típicos autómatas que siempre resultan aburridos; parece que con sólo mirarlos fijamente salgan a escape, mientras que si no tenemos balas se percatan de nuestro miedo y nos atacan.

Todos los camarotes están perfectamente ambientados, con cantidad de objetos, burbujas que suben constantemente, etc. Puede ser bastante útil dibujar un mapa para no perdernos con tanta habitación, aunque los objetos en ocasiones cambian de sitio y no hay quién los localice.

ADICCION:	△△△△△△△△
PRESENTACION:	△△△△△△△
GRAFICOS:	△△△△△△△
ACCION:	△△△△△△△△

Los efectos sonoros cumplen su misión perfectamente. Además al comienzo de cada juego podemos disfrutar de una bellísima melodía del más puro estilo barroco. Las arpas que podemos encontrar por el barco tienen la curiosa facultad de tocar una corta cancioncilla cada vez que las cogemos.

Resumiendo, que este es un juego en el que tanto la estrategia como la acción son necesarias si queremos llegar a algo. Un juego entretenido, con un argumento interesante y el nivel de dificultad justo para hacerlo atractivo desde el principio sin agotarlo en el primer día.



Programa: Cylu
Tipo: juego
Distribuidor: Serma
Formato: cassette
Precio: 1.200 ptas.





Nuestra tarea comienza en una pequeña y atestada habitación en lo más profundo de un gran laberinto. Tenemos bajo nuestro mando a Cylu, un pequeño robot que debemos guiar a lo largo y ancho del laberinto. Ocultos en éste hay 24 objetos que debemos devolver al ordenador principal. Durante la misión encontraremos 32 campos de fuerza que nos impedirán el paso a muchas secciones del laberinto. Encontraremos objetos curiosos, como CPUs, que podremos transportar (hasta un máximo de cinco), utilizar (los utilizables), examinar o volver a dejarlos.

En la parte baja de la pantalla aparecen los objetos que vayamos recogiendo, sobre alguno de ellos habrá una llave que indica que es el seleccionado (con SHIFT) sobre el que actuarán todas las funciones. Hay tanques de combustible que debemos utilizar cuando nos haga falta (también en la parte baja hay un contador que nos lo indica).

El laberinto, en tres dimensiones, es grande y complejo, y además sólo podemos ver una pequeña parte de él, por lo que es conveniente hacer un mapa según avanzamos.

Los gráficos son buenos, pero al ser muy pequeños, hallarse apiñados y medio tapados por partes del laberinto, no siempre son reconocibles. Los efectos sonoros son bastante buenos, como también lo es la melodía inicial. Aunque no es excesivamente lento para el tipo de gráficos que incluye, acusa una falta de acción bastante grande. Es más un juego de estrategia y aventura que un arcade típico.

Adicción: △△△△△△△
Presentación: △△△△△
Gráficos: △△△△△△△△
Acción: △△△△△△

DUMMY RUN

Programa: Dummy Run
Tipo: juego
Distribuidor: Erbe
Formato: cassette



HERBERT, el incansable bebé de «Everyone's A Wally», sigue haciendo de las suyas. Ha crecido y se ha convertido en un travieso niño que corretea y salta sin estarse quieto ni un momento. En esta ocasión ha ido de compras con Wally y Wilma y se ha perdido en unos grandes almacenes. Nuestra misión es guiarle a través de las secciones de la tienda y encontrar, antes de la hora de cierre, la sala de objetos perdidos donde Wally y Wilma le esperan desconsolados.

La estructura del juego es bastante clásica, diversas habitaciones comunicadas entre sí en las que hay distintos tipos de enemigos. Estos disminuirán nuestra energía hasta hacernos coger buenas pataletas, o, a la tercera de éstas, finalizar el juego. Para recuperarnos de esto debemos comer todas las golosinas que encontremos.

Algunas habitaciones toman la forma de pequeños juegos de arcade. Esta estrategia, cada vez más utilizada en programas de este tipo, es muy de agradecer, pues corta el ritmo del juego y lo hace mucho más entretenido.

Diseminados por las habitaciones hay objetos que nos resultarán útiles para fines concretos. Por ejemplo, la raqueta sirve para jugar al tenis, la bomba para destruir los edificios, y la llave para abrir la caja sorpresa y, aprovechando el impulso que ésta nos da, subir a los estantes de más arriba. Para coger los objetos (máximo dos), basta con pasar sobre ellos, automáticamente se toma el que había y se deja el penúltimo que cogimos.

Hay cuatro plantas distintas. Para cambiar de planta podemos elegir entre hacerlo por las escaleras o usar el ascensor. Si elegimos este medio hay que tener mucho cuidado de que el ascensor se encuentre en la planta en que estamos, pues si no caeríamos por el hueco. Aunque disponemos de un paracaídas para estos casos, no es conveniente abusar, pues se pierde bastante energía en las caídas.

Los gráficos del juego son muy variados y bastante cuidados. Las habitaciones están llenas de objetos, diferentes en cada una de ellas, y los personajes también son variados. El movimiento es bueno, aunque en los diversos enemigos excesivamente automatizado. La presentación general más que aceptable, siguiendo la tónica a que nos tiene acostumbrados Micro-Gen.

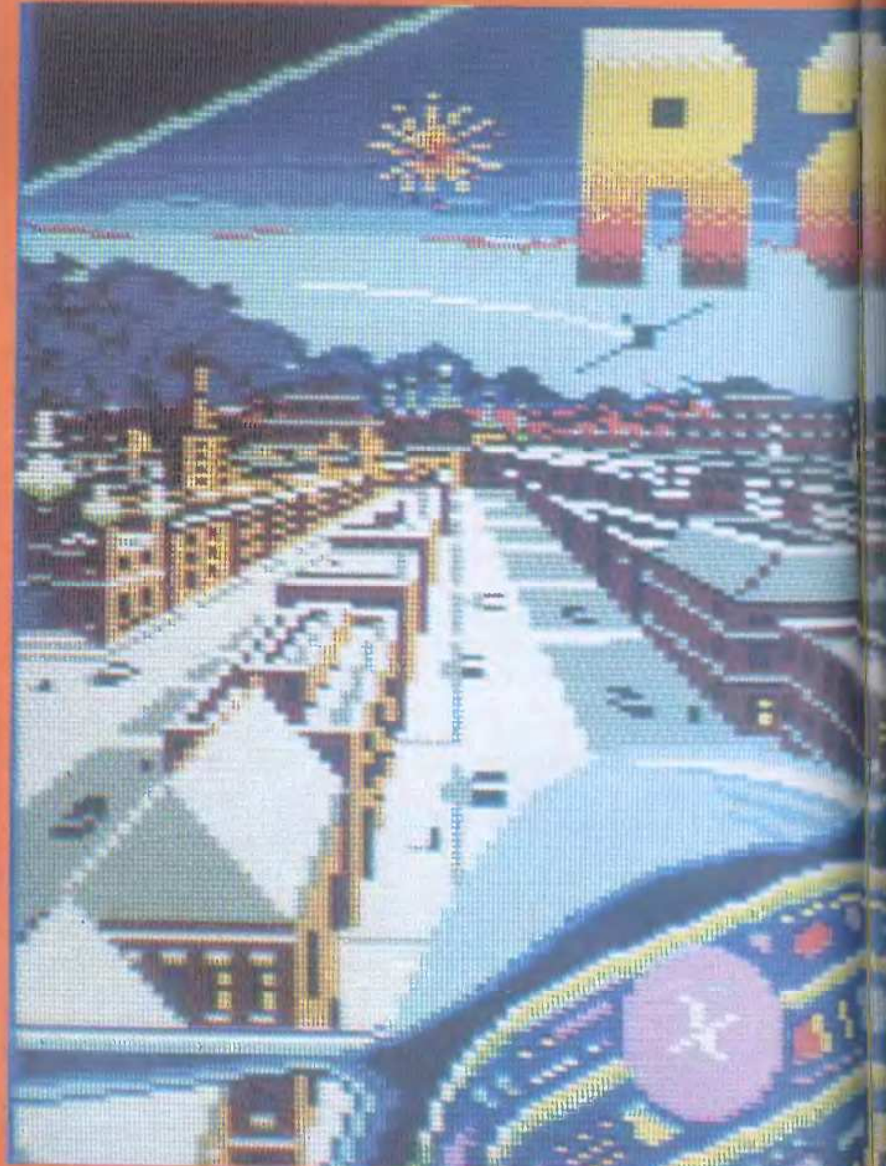
ADICCION: △△△△△△△
PRESENTACION: △△△△△△△△
GRAFICOS: △△△△△△△△
ACCION: △△△△△△△

Nos encontramos en esta ocasión con un programa que, sin salirse de la categoría de los arcades y juegos de acción, aporta nuevos planteamientos para este tipo de juegos. Normalmente los programadores crean una serie de rutinas de movimiento y una estructura, más o menos complicada, que se repite a medida que se desarrolla el juego. Pueden cambiar escenarios, dificultad o las características de los enemigos, pero normalmente la estructura general permanece de principio a fin. En el caso que nos ocupa, según avanzamos en el juego, encontramos una serie de pantallas, que, aunque unidas por una línea argumental y de gráficos bastante continuada, parece que fueran de programas distintos.

Nuestro objetivo: destruir las bases soviéticas de lanzamiento de misiles para así salvar al mundo (entiéndase EE.UU.) de un desastre nuclear. Toda una misión suicida, hay que cruzar el espacio aéreo ruso, destruir las bases antes que los misiles de éstas hagan explosión, y, ya que estamos, atacar el Centro de Defensa en el mismo Moscú y destruir su reactor nuclear. Para hacer todo esto hay que pasar por muchas fases distintas.

Programa: RAID!!!
Tipo: juego
Distribuidor: Erbe
Formato: cassette
Precio:

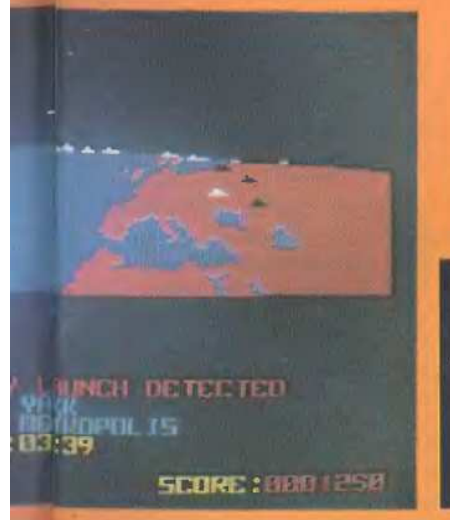




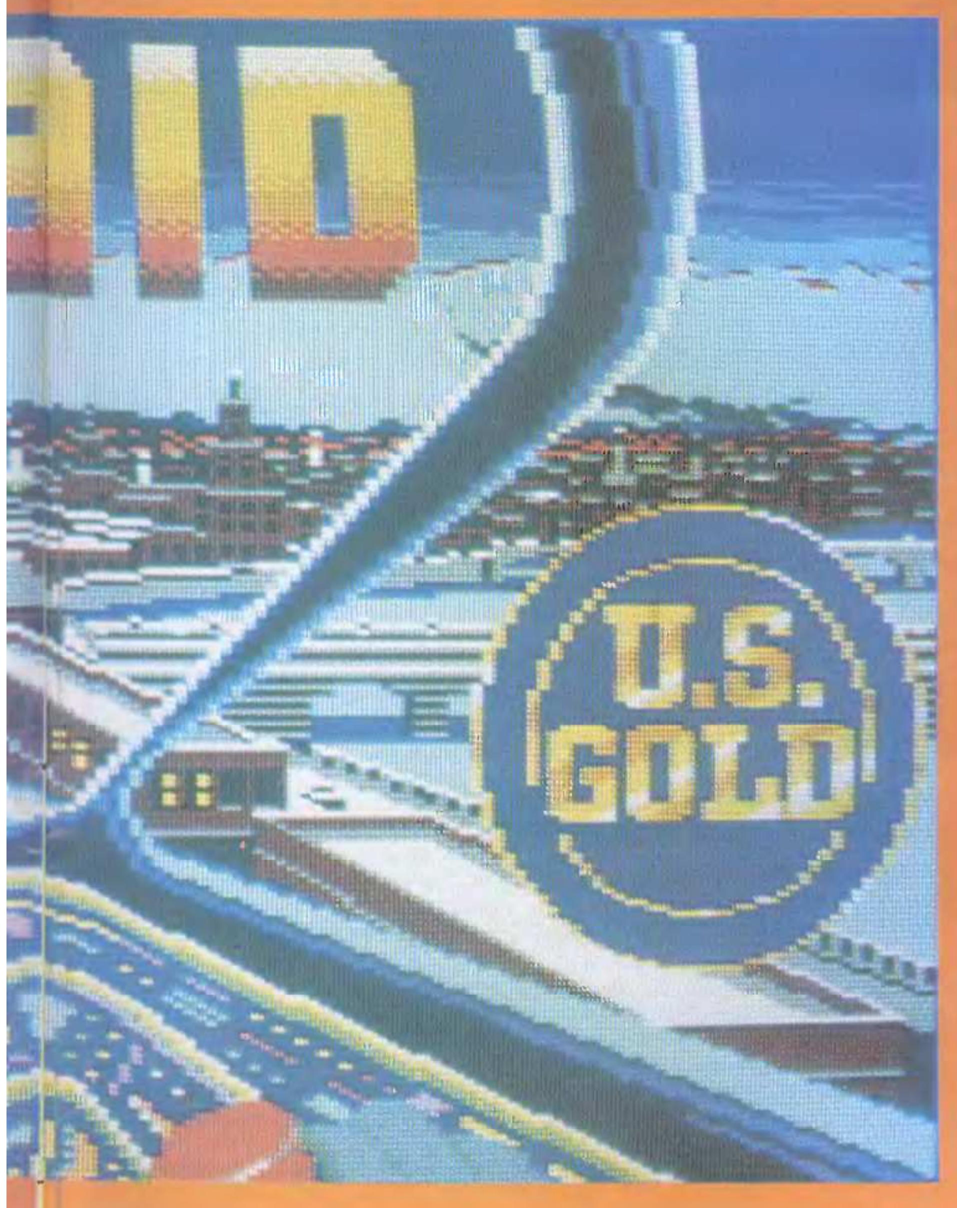
El cuartel general del SAC

Comenzamos con una vista del radar del SAC (Cuartel General de la Aviación Estratégica USA). Con él tendremos una visión panorámica del hemisferio septentrional, en donde se pueden distinguir claramente Europa, EE.UU., la posición de naves y misiles, y la situación en que se encuentran las bases rusas y americanas. También podremos ver, flotando sobre la Tierra, la estación espacial, equipada con todo un escuadrón de cazabombarderos Stealth. Un marcador en la parte inferior de la pantalla nos indicará el tiempo que queda antes de que los misiles enemigos lleguen a su destino; este dato puede sernos de gran utilidad, como veremos más adelante.

Cuando detectemos un lanza-



ADICCION:	△△△△△△△
DIFICULTAD:	△△△△△△△△△
GRAFICOS:	△△△△△△△
ACCION:	△△△△△△△△



miento por parte de los rusos, podremos pasar, pulsando el botón de disparo, a la estación espacial para hacernos cargo de la situación. Es aquí cuando comienzan los problemas.

En el hangar

Un vez dentro de la estación, nos encontramos con un hangar en el que descansan las 10 naves que debemos utilizar para penetrar en el espacio aéreo enemigo. Como comandantes del escuadrón somos los responsables de sacar las naves del hangar en buenas condiciones. Esto, que puede parecer fácil a primera vista, es uno de los pasos más complicados que habremos de seguir si queremos finalizar el juego.

El botón de disparo hará que nos elevemos del suelo, debe pulsarse de vez en cuando, pues, aunque la gravedad es muy leve, no pasa mucho tiempo sin que volvamos a caer. Empujando el mando hacia delante encendemos el motor principal, lo cual aumentará la velocidad en la dirección a que apuntemos. Empujando a derecha o izquierda haremos que gire la nave en esa dirección, pero no avanzará hacia ella si no encendemos el motor principal después de esto.

La inercia que toma el avión es tal que en los primeros intentos es casi imposible hacerse con el control del aparato, con resultados poco esperanzadores para el jugador. Además, una vez que conseguimos mediomantener bajo control surge el problema de que la puerta parece no querer abrirse mientras no avancemos en dirección contraria a ella a la vez que nos elevamos, y por si fuera poco se abre el tiempo suficiente para «aguillotinarnos» justo cuando vamos a pasar.

No hay que abdicar, pues con el tiempo las cosas van yendo mejor. Incluso puede que de vez en cuando logremos sacar algún que otro avión del hangar y podamos pasar a la siguiente fase del juego.

Hacia las bases enemigas

Una vez que acabamos con todo el escuadrón, aparece de nuevo la



vista general en la que los cazas que hayamos podido sacar estarán simbolizados por pequeños puntos blancos junto a la estación espacial. Uno de ellos parpadeará señalando que es el que debemos guiar hasta su destino: la base soviética que haya lanzado el misil, la cual aparecerá en blanco para destacarse de las demás.

Una vez que llegemos a ella habrá que cruzar una zona de terreno enemigo antes de poder destruirla. Aquí los mandos de control se parecen más a lo que estamos acostumbrados, pero el ver a la nave desde un lado confunde al principio.

Conviene volar bajo para despistar al radar y evitar los misiles antiaéreos, pero hay que tener cuidado con los obstáculos que vamos encontrando, y especialmente con las armas defensivas, que deberemos destruir a tiempo.

Si cruzamos esta zona con éxito pasaremos a una nueva fase en la que debemos destruir los silos de la base de lanzamiento de misiles. El control aquí es parecido al anterior, pero la perspectiva totalmente distinta. Conviene estar atento al tiempo que queda para que nuestras bases sean destruidas, y si éste es pequeño destruir el silo principal, que es el que controla todo.

Hay que tener cuidado con las naves enemigas que aparecen por la izquierda, y deshacernos de ellas antes de que bajen demasiado. Si conseguimos destruir las tres bases defensivas echaremos pie a tierra para pasar a una nueva fase en la que intentaremos destruir el centro de defensa soviético.



Más cerca del objetivo

En una trinchera frente al centro de defensa, y armado con un bazoka, estás solo ante el peligro. Puedes desplazarte de derecha a izquierda y controlar el ángulo de tiro de tu arma. Habrá soldados en los muros a ambos lados que irán a por ti, y tanques que saldrán de las puertas laterales. Pero el objetivo principal es la puerta del reactor nuclear, una cada vez, que reconocerás porque se pone blanca cuando se le dispara (lo cual puede ocurrirle a cualquiera).

Una vez destruida esta puerta y cuando no queden tanques ni soldados a la vista, podrás pasar a la siguiente fase, en la que tu misión es hacer que explote el reactor destruyendo a los robots de mantenimiento y evitando que puedan refrigerarlo. Los robots no son tontos, una vez que te localicen intentarán destruirte, por lo que debes anticiparte a ellos y dispararles por la espalda, que es su único punto débil.

Dispones de unas granadas en forma de discos que puedes enviarle «de rebote» contra la pared posterior. Para ayudarte en esta tarea, un sistema de guía por láser marcará un punto negro en la pared, que podrás desplazar dando adelante y atrás y alinear con los robots antes de enviarles las granadas.

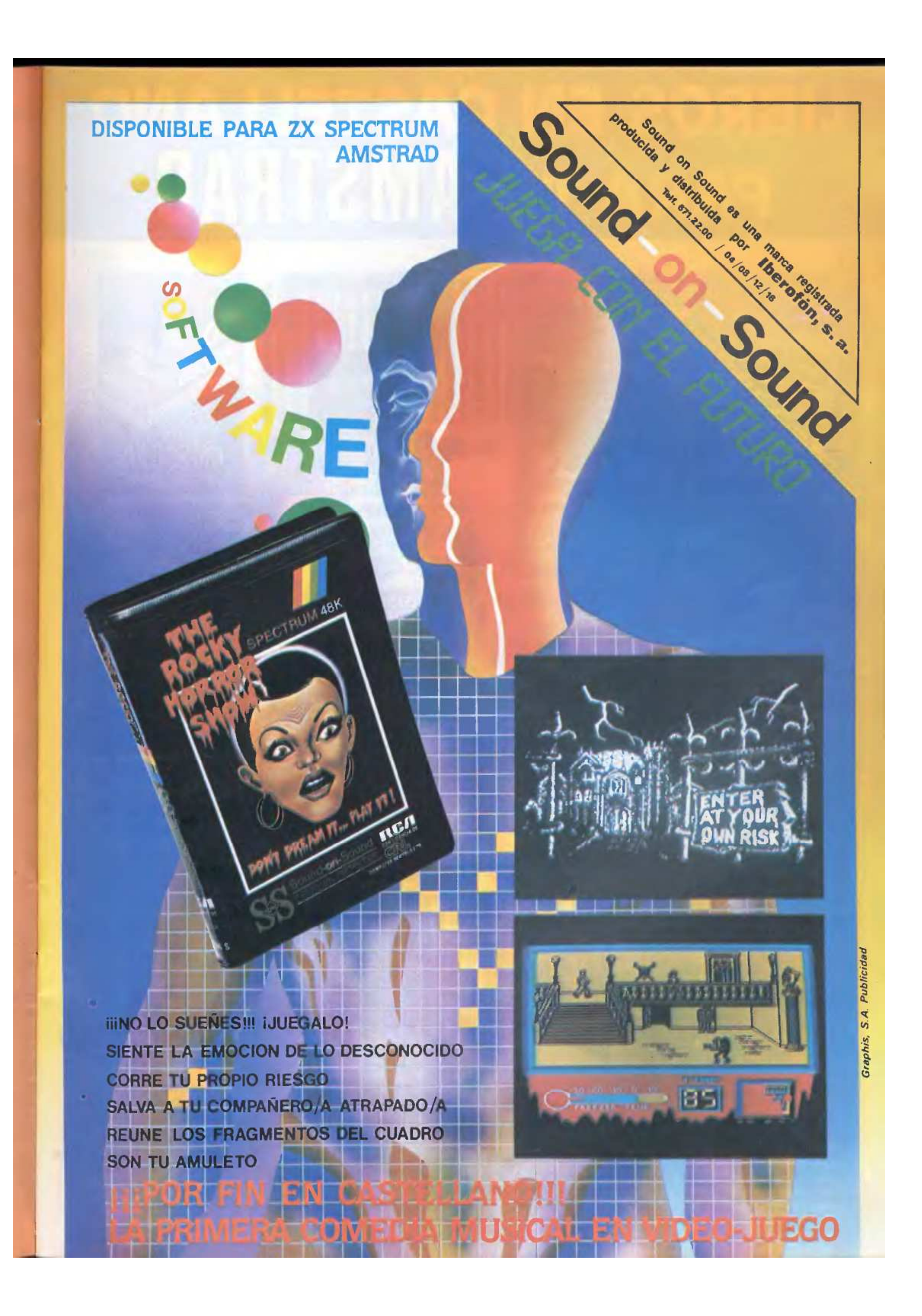
Si consigues destruir los robots debes escapar de allí antes de que el reactor haga explosión. Lo que sucede después dejaremos que lo descubras por ti mismo.

Un trabajo made in USA

Como en casi todo lo que se hace en América, al «destripar» este programa queda patente la profesionalidad de sus autores. Originario del Commodore, y, aunque algo «recortado» en su versión Amstrad, hace que nos demos cuenta de lo que pueden dar de sí 64K en un microordenador.

El movimiento es impecable a lo largo de todo el juego, y el efecto tridimensional está suficientemente conseguido gracias a las sombras que siguen a cada una de las naves. En los gráficos no se ha buscado la complejidad que hace que muchos de los últimos programas queden como faltos de «chispa», pero tienen estilo y cumplen perfectamente su papel. El nivel de dificultad es muy elevado, no es de los programas con los que se acaba el primer día ni el segundo, pero gracias a la variedad en las pantallas esto no lo hace aburrido.

En definitiva, un gran juego para quien le guste la máxima dificultad y tenga aficiones bélicas contra los rusos.


DISPONIBLE PARA ZX SPECTRUM
AMSTRAD
SOFTWARE
Sound-on-Sound
Sound on Sound es una marca registrada
producida y distribuida por Iberofón, s. a.
Telf. 671.22.00 / 04/08/12/16
JUEGA CON EL FUTURO
THE ROCKY HORROR SHOW
SPECTRUM 48K
DON'T DREAM IT... PLAY IT!
RCA
ENTER AT YOUR OWN RISK
¡¡¡NO LO SUEÑES!!! ¡JUEGALO!
SIENTE LA EMOCION DE LO DESCONOCIDO
CORRE TU PROPIO RIESGO
SALVA A TU COMPAÑERO/A ATRAPADO/A
REUNE LOS FRAGMENTOS DEL CUADRO
SON TU AMULETO
¡¡¡POR FIN EN CASTELLANO!!!
LA PRIMERA COMEDIA MUSICAL EN VIDEO-JUEGO
Graphis, S.A. Publicidad

LIBROS EN CASTELLANO PARA TU AMSTRAD



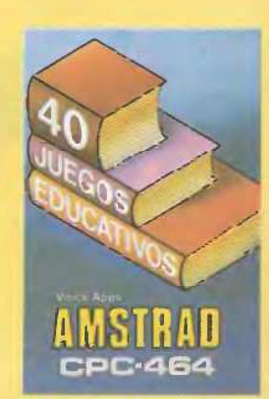
Manual de Referencia Basic para el Programador
La más autorizada y completa guía para programar en Locomotive Basic.
3.400.— Pts.



Juegos Sensacionales para AMSTRAD
Listados completos de 27 estupendos juegos de muy diversos estilos.
1.950.— Pts.



Programando con AMSTRAD
Fundamental para el usuario principiante.
Ameno y repleto de ejemplos.
2.400.— Pts.



40 Juegos Educativos
Listados completos (matemáticas, geografía, música, etc.) para aprender divirtiéndose.
1.950.— Pts.



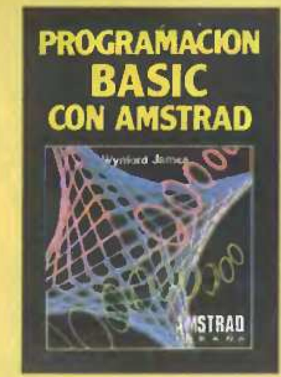
Código máquina para principiantes con AMSTRAD
Ideal para iniciarse en el código máquina del Z80 y en el sistema operativo del AMSTRAD.
2.100.— Pts.



Hacia la Inteligencia Artificial con AMSTRAD
Convierta su AMSTRAD en un compañero inteligente
1.500.— Pts.



Música y Sonidos con AMSTRAD
Programe música y efectos sonoros y convierta su AMSTRAD en un sintetizador.
1.200.— Pts.



Programación de Basic con AMSTRAD.
Imprescindible para el principiante y eficaz herramienta para el programador avanzado.
2.100.— Pts.



Técnicas de Programación de Gráficos en el AMSTRAD
Este libro enseña a aprovechar las excelentes funciones gráficas del AMSTRAD, con múltiples ejemplos.
1.950.— Pts.



Curso Autodidáctico de Basic I y II
Un completo y estructurado Curso de Basic apoyado con numerosos ejemplos y acompañado de cassettes.
2.900.— Pts. cada volumen

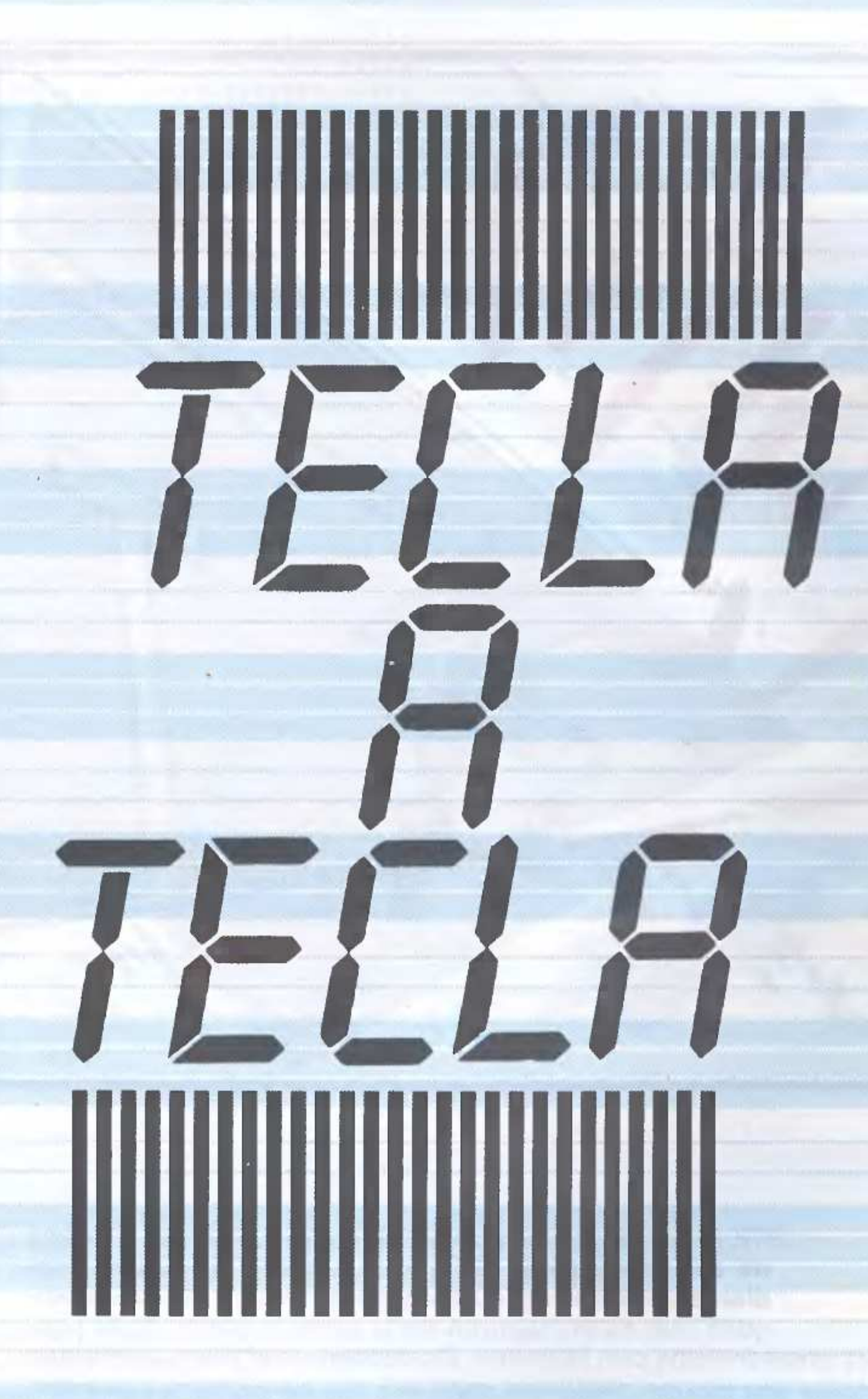


Avd. del Mediterráneo, 9
Telfs.: 433 45 48 — 433 48 76
28007 MADRID

Delegación en Cataluña:
C/. Tarragona, 110 — Telf. 325 10 58
08015 BARCELONA

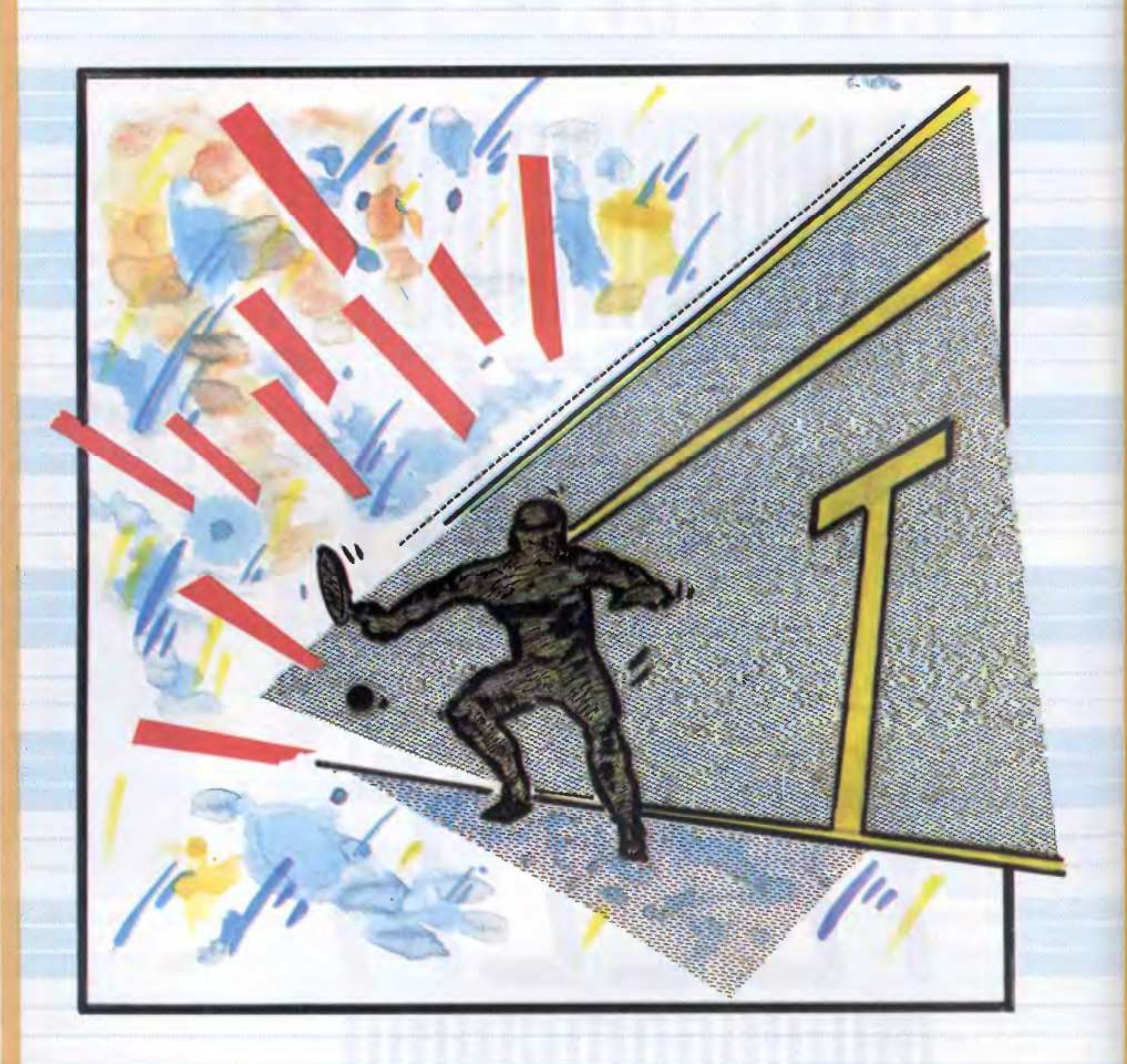
DE VENTA EN EL CORTE INGLES
Y TIENDAS ESPECIALIZADAS

Marca Registrada por el Grupo Indescomp.



PROGRAMAS

FRONTON



ESTE es un programa muy sencillo aunque no por ello deja de ser entretenido. El jugador controla la raqueta mediante las teclas del cursor y debe destruir todos los bloques que se encuentran en la parte superior de la pantalla golpeándolos con la pelota. Comenzamos el juego con cinco pelotas que iremos perdiendo cada vez que no logremos golpearla con la raqueta. Aun siendo un programa muy simple la dificultad del juego es bastante elevada siendo muy difícil concluir una partida con éxito.

```
10 REM ********* FRONTON ***********
20 REM ***** AMSTRAD USER 1986 *****
30 REM ****************************
40 REM
50 BORDER 1:INK 0,1:INK 2,24:INK 3,6
60 SPEED KEY 15,2
70 ENV 1,1,18,0,11,0,10:ENT 1,10,2,2
80 ENV 3,1,0,16,5,-3,2
90 ENV 2,5,3,3,1,-21,22,9,-3,2:ENT -2,10,2,2,5,-7,1,2,11,3,2,-4,8
100 MODE 1
110 MOVE 30,16:DRAWR 0,400,1:MOVE 610,16:DRAWR 0,400,1
120 PEN 3:LOCATE 3,1:PRINT STRING$(36,143) :PEN 2:LOCATE 3,2:PRINT STRING$(36,14
3):PEN 1:FOR r=5 TO 6:LOCATE 3,r:PRINT STRING$(36,14
3):NEXT r
130 bx=9
140 VID=5:PUN=0
150 PEN 1:GOSUB 460
160 IF INKEY$<>" "THEN 160
170 GOTO 200:REM comienzo del juego
180 LOCATE bx,24:PRINT"  ";STRING$(4,131);"  "
190 RETURN
200 xa=1:ya=1:IF INT(RND*2)=1 THEN xa=-xa
210 PEN 1:GOSUB 180
220 ORIGIN 0,400
230 x=bx+4:y=11:x1=x:y1=y
240 x1=x+xa:y1=y+ya:REM comienzo del bucle principal
250 IF x1=3 OR x1=38 THEN xa=-xa
260 GOSUB 400
270 IF y1=24 AND x1>bx+1 AND x1<bx+6 THEN ya=-ya:y1=y1-2:SOUND 130,44,6,7,1,1:a=
((X>bx+5)OR(x<bx+2)):IF a=-1 THEN xa=xa*a:x1=x1+xa:y
1=y1+1
280 IF y1=25 THEN LOCATE x,y:PRINT" ":GOTO 370
290 GOSUB 180
300 t=TEST ((16*x1)-1,-(16*y1)-1):IF t<>0 THEN ya=-ya:xz=x1:yz=y1:y1=y1+ya:GOSUB
 430:IF t=2 THEN pun=pun+10:GOSUB 460
310 IF t=3 THEN pun=pun+20:GOSUB 460
320 IF t=1 THEN pun=pun-5:GOSUB 460
330 IF y1=1 THEN ya=1
340 LOCATE x,y:PRINT " ":LOCATE x1,y1:PRINT CHR$(233):x=x1:y=y1
350 IF y=1 OR x=3 OR x=38 THEN SOUND 129,78,6,7,1,1
360 GOTO 240:REM fin del bucle principal
370 vid=vid-1:SOUND 172,19,46,12,2,2:IF vid=0 THEN GOTO 440
380 GOSUB 460
390 GOTO 200
400 IF (INKEY(8)=0 OR INKEY(74)=0)AND BX>2 THEN BX=BX-2:RETURN
410 IF (INKEY(1)=0 OR INKEY(75)=0)AND bx<32 THEN bx=bx+2:RETURN
420 RETURN
430 LOCATE xz,yz:PRINT " ":RETURN
440 IF pun>=rec THEN rec=pun
450 pun=0:vid=5:GOSUB 460:GOTO 100
460 SOUND 130,0,20,17,3,0,31:LOCATE 1,25:PRINT"RECORD ";rec;:LOCATE 16,25:PRINT"
PUNTOS ";pun:LOCATE 30,25:PRINT"VIDAS ";vid:RETURN
```



OTHELO

ESTE interesante juego no es muy conocido en España. Se desarrolla en un tablero similar al del ajedrez, consiste en capturar entre dos fichas propias el mayor número posible de fichas del contrario volviéndose entonces de nuestro color. Podemos elegir el color de nuestras fichas (blancas o negras), teniendo en cuenta que las fichas negras siempre mueven primero. Para realizar las jugadas debemos introducir la fila y la columna a donde deseamos movernos. Los movimientos ilegales son rechazados con un mensaje de advertencia debiendo repetir la jugada. Gana el jugador que tenga más fichas al final del juego. El programa es bastante bueno, con un nivel de juego alto, lo que garantiza muchas horas de dura competencia. Solamente se echa en falta una opción que permita dosificar el nivel de juego y la posibilidad de juego contra otra persona.

```
10 REM **** OTHELO POR  M.J.GRIBBI ****
20 REM ******* AMSTRAD USER 1986 ******
30 REM
40 BORDER 14
50 CLEAR
60 MODE 1:PEN 0:PAPER 1:CLS
70 INK 0,0:INK 1,14:INK 2,18:INK 3,26
80 LOCATE 2,3:PEN 3:PRINT"O":LOCATE 3,4:PRINT"T":LOCATE 4,5:PRINT"H":LOCATE 5,6:
PRINT"E"
90 LOCATE 6,7:PRINT"L":LOCATE 7,8:PRINT"O"
100 WINDOW #1,2,39,22,25:PAPER #1,1:PEN #1,0:CLS #1
110 PEN 0
120 LOCATE #1,13,1:PRINT #1,"  LAS NEGRAS SIEMPRE JUEGAN PRIMERO"
130 LOCATE #1,1,3:PRINT #1,"PULSE N O B PARA ELEGIR NEGRO O BLANCO"
140 B$=INKEY$:IF B$="" THEN 140
150 IF B$="B" OR  B$="b" THEN Q%-3:N%=0:GOTO 210
160 IF B$="N" OR  B$="n" THEN Q%=0:N%=3:GOTO 210
170 CLS #1:LOCATE #1,4,3
180 PRINT #1,"        SOLO BLANCAS O NEGRAS"
190 FOR T=0 TO 1000:NEXT T
200 GOTO 140
210 DIM C%(10,10),P%(9,9),C1%(8),C2%(8),CX%(9),CY%(9)
220 I1%=2:J1%=2:I2%=7:J2%=7
230 FOR I%=0 TO 9
240 C%(I%,0%)=6:C%(0,I%)=6
250 C%(9,I%)=6:C%(I%,9)=6
260 NEXT I%
270 FOR I%=1 TO 8
280 READ C1%(I%),C2%(I%)
290 FOR J%= 1 TO 8
300 READ P%(I%,J%)
310 C%(I%,J%)=6
320 NEXT J%:NEXT I%
330 C%(4,4)=3:C%(4,5)=0:C%(5,4)=0:C%(5,5)=3
340 FOR K%=1 TO 58
350 READ AR%,BR%,CR%,DR%
360 PLOT AR%,BR%:DRAW CR%,DR%,0
370 NEXT K%
380 GOSUB 1460
390 IF Q%=3 GOTO 770
400 CLS #1:INPUT #1,"  A QUE FILA MUEVES";E%
410 IF E% <1 OR E% >8 GOTO 400
420 LOCATE #1,1,3:INPUT #1,"   A QUE COLUMNA ";D%
430 IF D% <1 OR D% >8 GOTO 420
440 IF C%(D%,E%)=6 GOTO 480
450 CLS #1:LOCATE #1,5,2:PRINT #1," ESTA POSICION ESTA OCUPADA !"
460 FOR T=1 TO 1000:NEXT T
470 GOTO 400
480 PLOT 270+(30*D%),70+(30*E%):DRAW 290+(30*D%),89+(30*E%),Q%
490 PLOT 290+(30*D%),70+(30*E%):DRAW 270+(30*D%),89+(30*E%),Q%
500 GOTO 540
510 FOR M%= 0 TO 19 STEP 2:PLOT 270+(30*D%),70+M%+(30*E%)
520 DRAW 290+(30*D%),70+M%+(30*E%),6:NEXT M%
530 GOTO 400
540 VRX%=0
550 FOR K%=1 TO 8
560 VR%=0:C3%=D%:C4%=E%
570 C3%=C3%+C1%(K%):C4%=C4%+C2%(K%)
580 IF C%(C3%,C4%)=N% GOTO 590 ELSE 600
```

PROGRAMAS

```
590 VR%=VR%+1:GOTO 570
600 IF C%(C3%,C4%)=6 GOTO 610 ELSE 620
610 NEXT K%:GOTO 670
620 IF VR%=0 GOTO 610 ELSE 630
630 VRX%=VRX%+VR%
640 C3%=C3%-C1%(K%):C4%=C4%-C2%(K%)
650 IF C%(C3%,C4%)=6 GOTO 610 ELSE 660
660 C%(C3%,C4%)=Q%:GOTO 640
670 IF VRX%=0 GOTO 680 ELSE 710
680 CLS #1:PRINT #1," ESA JUGADA NO ES VALIDA"
690 FOR T=1 TO 1000:NEXT T
700 GOTO 510
710 E%=E%:D%=D%:VRX%=VRX%
720 CLS #1:PRINT #1,"HAS MOVIDO A LA FILA NUMERO ";E%
730 PRINT #1,"          Y A LA COLUNNA NUMERO ";D%
740 LOCATE #1,2,4:PRINT #1,"ESTO TE DA ";VRX%;"  CUADRO(S)"
750 C%(D%,E%)=Q%:GOSUB 1710
760 GOSUB 1460
770 CLS #1:LOCATE #1,10,2:PRINT #1,"ESTE ES MI TURNO...!"
780 P%=0:VRX%=0:VRY%=0
790 IF I1%*J1%=1 AND I2%*J2%=64 GOTO 860
800 FOR K%=2 TO 7
810 IF C%(2,K%) <> 6 THEN I1%=1
820 IF C%(7,K%) <> 6 THEN I2%=8
830 IF C%(K%,2) <> 6 THEN J1%=1
840 IF C%(K%,7) <> 6 THEN J2%=8
850 NEXT K%
860 FOR I%=I1% TO I2%
870 FOR J%=J1% TO J2%
880 IF C%(I%,J%)=6 GOTO 1030
890 NEXT J%:NEXT I%
900 IF P% > 0 GOTO 1000
910 IF PAS%=1 GOTO 920 ELSE 940
920 CLS #1:PRINT #1," BLOQUEADO ! YO DEBO FINALIZAR"
930 FOR T=1 TO 1000:NEXT T:GOTO 1550
940 CLS #1:LOCATE #1,18,2:PRINT #1,"I MUST PASS"
950 GOSUB 2720
960 IF PAS%=1 GOTO 970 ELSE 990
970 CLS #1:PRINT #1,"BLOQUEADO ! YO DEBO FINALIZAR."
980 FOR T=1 TO 1000:NEXT T:GOTO 1550
990 GOTO 400
1000 IF LC%=0 THEN LC%=1:RANDOMIZE LC%:RL%=RND(LC%)
1010 CX1%=CX%(RL%):CX2%=CY%(RL%)
1020 GOTO 1220
1030 VRX%=0
1040 FOR K%=1 TO 8
1050 VR%=0:C3%=I%:C4%=J%
1060 C3%=C3%+C1%(K%):C4%=C4%+C2%(K%)
1070 IF C%(C3%,C4%)=Q% GOTO 1080 ELSE 1090
1080 VR%=VR%+1:GOTO 1060
1090 IF C%(C3%,C4%)=6 GOTO 1100 ELSE 1110
1100 NEXT K%:GOTO 1130
1110 IF VR%=0% GOTO 1100 ELSE 1120
1120 VRX%=VRX%+VR%:GOTO 1100
1130 IF VRX%=0 GOTO 890
1140 IF P%(I%,J%) < P% GOTO 890
1150 IF P%(I%,J%) > P% GOTO 1160 ELSE 1170
1160 P%=P%(I%,J%):VRY%=VRX%:LC%=0:CX%(0)=I%:CY%(0)=J%:GOTO 890
1170 IF VRY% > VRX% GOTO 890
1180 IF VRY% < VRX% GOTO 1190 ELSE 1200
1190 LC%=0:VRY%=VRX%:CX%(0)=I%:CY%(0)=J%:GOTO 890
1200 LC%=LC%+1:CX%(LC%)=I%:CY%(LC%)=J%
```

```
1210 GOTO 890
1220 CX2%=CX2%:CX1%=CX1%:VRY%=VRY%
1230 CLS #1:PRINT #1," YO ELIJO LA LINEA NUMERO ";CX2%
1240 PRINT #1,"        Y LA COLUMNA NUMERO ";CX1%
1250 LOCATE #1,1,4:PRINT #1,"ESTO ME DA ";VRY%;" CUADRO(S)"
1260 PLOT 270+(30*CX1%),70+(30*CX2%):DRAW 290+(30*CX1%),89+(30*CX2%),N%
1270 PLOT 290+(30*CX1%),70+(30*CX2%):DRAW 270+(30*CX1%),89+(30*CX2%),N%
1280 FOR T=1 TO 1000:NEXT T
1290 FOR K%=1 TO 8
1300 VR%=0:C3%=CX1%:C4%=CX2%
1310 C3%=C3%+C1%(K%):C4%=C4%+C2%(K%)
1320 IF C%(C3%,C4%)=Q% GOTO 1330 ELSE 1340
1330 VR%=VR%+1:GOTO 1310
1340 IF C%(C3%,C4%)=6 GOTO 1350 ELSE 1360
1350 NEXT K%:GOTO 1400
1360 IF VR%=0 GOTO 1350
1370 C3%=C3%-C1%(K%):C4%=C4%-C2%(K%)
1380 IF C%(C3%,C4%)=6 GOTO 1350
1390 C%(C3%,C4%)=N%:GOTO 1370
1400 C%(CX1%,CX2%)=N%
1410 GOSUB 2720
1420 GOSUB 1460
1430 IF PAS%=1 GOTO 1440 ELSE 1450
1440 CLS #1:PRINT #1,"      DEBES PASAR":FOR T=1 TO 1000:NEXT T:GOTO 770
1450 GOTO 400
1460 FOR I%=1 TO 8
1470 FOR J%=1 TO 8
1480 FOR M%=0 TO 19 STEP 2
1490 Z%=270+(30*I%):H%=70+(30*J%):W%=H%+M%
1500 PLOT Z%,W%:DRAW Z%+20,W%,C%(I%,J%)
1510 NEXT M%:NEXT J%:NEXT I%
1520 X%=X%+1
1530 IF X%=61 GOTO 1550
1540 RETURN
1550 CQ%=0:CN%=0
1560 FOR I%=1 TO 8
1570 FOR J%=1 TO 8
1580 IF C%(I%,J%)=Q% THEN CQ%=CQ%+1
1590 IF C%(I%,J%)=N% THEN CN%=CN%+1
1600 NEXT J%:NEXT I%
1610 IF CQ% > CN% GOTO 1680
1620 IF CQ%=CN% GOTO 1630 ELSE 1650
1630 CLS #1:LOCATE #1,25,2:PRINT #1,"BLOQUEADO"
1640 END
1650 CLS #1:LOCATE #1,5,1:PRINT #1,"LLEVAS  ";CQ%;" CUADROS;YO LLEVO ";CN%
1660 LOCATE #1,10,3:PRINT #1,"HE GANADO...!!!!"
1670 END
1680 CLS #1:LOCATE #1,5,1:PRINT #1,"LLEVAS ";CQ%;" CUADROS;" YO LLEVO ";CN%
1690 LOCATE #1,5,3:PRINT #1,"BRAVO. HAS GANADO !!"
1700 END
1710 IF C%(2,2)=Q% AND (C%(3,1)=N% OR C%(1,3)=N%) GOTO 1720 ELSE 1730
1720 P%(3,1)=1:P%(1,3)=1
1730 IF C%(7,7)=Q% AND (C%(8,6)=N% OR C%(6,8)=N%) GOTO 1740 ELSE 1750
1740 P%(8,6)=1:P%(6,8)=1
1750 IF C%(2,7)=Q% AND (C%(1,6)=N% OR C%(3,8)=N%) GOTO 1760 ELSE 1770
1760 P%(1,6)=1:P%(3,8)=1
1770 IF C%(7,2)=Q% AND (C%(6,1)=N% OR C%(8,3)=N%) GOTO 1780 ELSE 1790
1780 P%(6,1)=1:P%(8,3)=1
1790 IF D%=1 OR D%=8 OR E%=1 OR E%=8 GOTO 1820
1800 IF CX1%=1 OR CX1%=8 OR CX2%=1 OR CX2%=8 GOTO 1820
1810 RETURN
1820 FOR J%=1 TO 8 STEP 7
```

```
1830 FOR I%=2 TO 7
1840 IF C%(I%,J%)=N% GOTO 1850 ELSE 1860
1850 P%(I%+1,J%)=21:P%(I%-1,J%)=21
1860 IF C%(J%,I%)=N% GOTO 1870 ELSE 1880
1870 P%(J%,I%+1)=21:P%(J%,I%-1)=21
1880 NEXT I%
1890 FOR I%=2 TO 7
1900 IF C%(I%,J%)=Q% GOTO 1910 ELSE 1920
1910 P%(I%+1,J%)=2:P%(I%-1,J%)=2
1920 IF C%(J%,I%)=Q% GOTO 1930 ELSE 1940
1930 P%(J%,I%+1)=2:P%(J%,I%-1)=2
1940 NEXT I%:NEXT J%
1950 P%(1,2)=1:P%(1,7)=1:P%(2,1)=1:P%(7,1)=1
1960 P%(2,8)=1:P%(7,8)=1:P%(8,2)=1:P%(8,7)=1
1970  FOR I%=2 TO 7
1980 IF C%(1,I%-1)=Q% AND C%(1,I%+1)=Q% THEN P%(1,I%)=25
1990 IF C%(8,I%-1)=Q% AND C%(8,I%+1)=Q% THEN P%(8,I%)=25
2000 IF C%(I%-1,1)=Q% AND C%(I%+1,1)=Q% THEN P%(I%,1)=25
2010 IF C%(I%-1,8)=Q% AND C%(I%+1,8)=Q% THEN P%(I%,8)=25
2020 NEXT I%
2030 FOR J%=1 TO 8 STEP 7
2040 FOR I%=4 TO 8
2050 IF C%(J%,I%) <> N% GOTO 2140
2060 IC%=I%-1:IF C%(J%,IC%)=6 GOTO 2140
2070 IF C%(J%,IC%)=Q% GOTO 2080 ELSE 2090
2080 IC%=IC%-1:GOTO 2070
2090 IF C%(J%,IC%)=6 GOTO 2110
2100 GOTO 2140
2110 IF IC%=0 GOTO 2140
2120 IF C%(J%,I%+1)=Q% AND C%(J%,IC%-1)=6 GOTO 2140
2130 P%(J%,IC%)=26
2140 IF C%(I%,J%) <> N% GOTO 2230
2150 IC%=I%-1:IF C%(IC%,J%)=6 GOTO 2230
2160 IF C%(IC%,J%)=Q% GOTO 2170 ELSE 2180
2170 IC%=IC%-1:GOTO 2160
2180 IF C%(IC%,J%)=6 GOTO 2200
2190 GOTO 2230
2200 IF IC%=0 GOTO 2230
2210 IF C%(I%+1,J%)=Q% AND C%(IC%-1,J%)=6 GOTO 2230
2220 P%(IC%,J%)=26
2230 NEXT I%
2240 FOR I%=1 TO 5
2250 IF C%(J%,I%) <> N% GOTO 2340
2260 IC%=I%+1:IF C%(J%,IC%)=6 GOTO 2340
2270 IF C%(J%,IC%)=Q% GOTO 2280 ELSE 2290
2280 IC%=IC%+1:GOTO 2270
2290 IF C%(J%,IC%)=6 GOTO 2310
2300 GOTO 2340
2310 IF IC%=9 GOTO 2340
2320 IF C%(J%,I%-1)=Q% AND C%(J%,IC%+1)=6 GOTO 2340
2330 P%(J%,IC%)=26
2340 IF C%(I%,J%) <> N% GOTO 2430
2350 IC%=I%+1:IF C%(IC%,J%)=6 GOTO 2430
2360 IF C%(IC%,J%)=Q% GOTO 2370 ELSE 2380
2370 IC%=IC%+1:GOTO 2360
2380 IF C%(IC%,J%)=6 GOTO 2400
2390 GOTO 2430
2400 IF IC%=9 GOTO 2430
2410 IF C%(I%-1,J%)=Q% AND C%(IC%+1,J%)=6 GOTO 2430
2420 P%(IC%,J%)=26
2430 NEXT I%:NEXT J%
2440 IF C%(1,1)=N% GOTO 2450 ELSE 2460
2450 FOR I%=2 TO 6:P%(1,I%)=20:P%(I%,1)=20:NEXT I%
```

```
2460 IF C%(1,8)=N% GOTO 2470 ELSE 2480
2470 FOR I%=2 TO 6:P%(I%,8)=20:P%(1,9-I%)=20:NEXT I%
2480 IF C%(8,1)=N% GOTO 2490 ELSE 2500
2490 FOR I%=2 TO 6:P%(9-I%,1)=20:P%(8,I%)=20:NEXT I%
2500 IF C%(8,8)=N% GOTO 2510 ELSE 2520
2510 FOR I%=3 TO 7:P%(I%,8)=20:P%(8,I%)=20:NEXT I%
2520 IF C%(1,1) <> 6 THEN P%(2,2)=5
2530 IF C%(1,8) <> 6 THEN P%(2,7)=5
2540 IF C%(8,1) <> 6 THEN P%(7,2)=5
2550 IF C%(8,8) <> 6 THEN P%(7,7)=5
2560 P%(1,1)=30:P%(1,8)=30:P%(8,1)=30:P%(8,8)=30
2570 FOR I%=3 TO 6
2580 IF C%(1,I%)=N% THEN P%(2,I%)=4
2590 IF C%(8,I%)=N% THEN P%(7,I%)=4
2600 IF C%(I%,1)=N% THEN P%(I%,2)=4
2610 IF C%(I%,8)=N% THEN P%(I%,7)=4
2620 NEXT I%
2630 IF C%(7,1)=Q% AND C%(4,1)=N% AND C%(6,1)=6 AND C%(5,1)=6 THEN P%(6,1)=26
2640 IF C%(1,7)=Q% AND C%(1,4)=N% AND C%(1,6)=6 AND C%(1,5)=6 THEN P%(1,6)=26
2650 IF C%(2,1)=Q% AND C%(5,1)=N% AND C%(3,1)=6 AND C%(4,1)=6 THEN P%(3,1)=26
2660 IF C%(1,2)=Q% AND C%(1,5)=N% AND C%(1,3)=6 AND C%(1,4)=6 THEN P%(1,3)=26
2670 IF C%(8,2)=Q% AND C%(8,5)=N% AND C%(8,3)=6 AND C%(8,4)=6 THEN P%(8,3)=26
2680 IF C%(2,8)=Q% AND C%(5,8)=N% AND C%(3,8)=6 AND C%(4,8)=6 THEN P%(3,8)=26
2690 IF C%(8,7)=Q% AND C%(8,4)=N% AND C%(8,5)=6 AND C%(8,6)=6 THEN P%(8,6)=26
2700 IF C%(7,8)=Q% AND C%(4,8)=N% AND C%(5,8)=6 AND C%(6,8)=6 THEN P%(6,8)=26
2710 RETURN
2720 PAS%=0
2730 FOR I%=1 TO 8
2740 FOR J%=1 TO 8
2750 IF C%(I%,J%)=Q% GOTO 2780
2760 NEXT J%:NEXT I%
2770 PAS%=1:RETURN
2780 FOR K%=1 TO 8
2790 VR%=0:C3%=I%:C4%=J%
2800 C3%=C3%+C1%(K%):C4%=C4%+C2%(K%)
2810 IF C3% < 1 OR C3% > 8 GOTO 2820 ELSE 2830
2820 NEXT K%:GOTO 2760
2830 IF C4% < 1 OR C4% > 8 GOTO 2820 ELSE 2840
2840 IF C%(C3%,C4%)=N% GOTO 2850 ELSE 2860
2850 VR%=VR%+1:GOTO 2800
2860 IF C%(C3%,C4%)=Q% GOTO 2820 ELSE 2870
2870 IF VR% > 0 THEN RETURN
2880 GOTO 2820
2890 DATA 1,0,30,1,20,10,10,20,1,30,1,1,1,1,3
2900 DATA 3,3,3,1,1,0,1,20,3,5,5,5,5,3,20,-1,1,10,3,5
2910 DATA 0,0,5,3,10,-1,0,10,3,5,0,0,5,3,10,-1
2920 DATA -1,20,3,5,5,5,5,3,20,0,-1,1,1,3,3,3,3,1,1,1,-1,30,1,20,10,10,20,1,30
2930 DATA 263,100,263,120,270,130,255,130,255,130,255,140,255,140,270,140
2940 DATA 270,140,270,150,270,150,255,150,255,160,270,160,270,160,270,180
2950 DATA 270,180,255,180,270,170,255,170,270,190,270,210,270,200,255,200
2960 DATA 255,200,255,210,255,220,270,220,270,220,270,230,270,230,255,230
2970 DATA 255,230,255,240,255,240,270,240,255,250,270,250,270,250,270,260
2980 DATA 270,260,255,260,255,250,255,270,270,280,270,300,270,300,255,300
2990 DATA 255,310,255,330,255,330,270,330,270,330,270,310,270,310,255,310
3000 DATA 255,320,270,320
3010 DATA 310,355,310,375,350,355,335,355,335,355,335,365,335,365,350,365
3020 DATA 350,365,350,375,350,375,335,375,365,355,380,355,380,355,380,375
3030 DATA 380,375,365,375,380,365,365,365,410,355,410,375,410,365,395,365
3040 DATA 395,365,395,375,425,355,440,355,440,355,440,365,440,365,425,365
3050 DATA 425,365,425,375,425,375,440,375,455,375,455,355,455,355,470,355
3060 DATA 470,355,470,365,470,365,455,365,485,375,500,375,500,375,500,355
3070 DATA 515,375,515,355,515,355,530,355,530,355,530,375,530,375,515,375
3080 DATA 515,365,530,365
```

PROGRAMAS

LUCHA ELECTRICA



EL objetivo de este juego para dos personas es el de hacer blanco sobre el contrario evitando al mismo tiempo ser alcanzado por sus disparos. El jugador que antes logre acertar nueve veces sobre el contrario se proclamará vencedor de la lucha. El control del programa se realiza mediante dos joysticks o teclado para esta opción las teclas utilizadas son las siguientes:

	JUGADOR 1	JUGADOR 2
ARRIBA:	A	F6
ABAJO:	Z	F3
DISPARO:	X	F7

```
10 REM ******  LUCHA ELECTRICA ******
20 REM ***** PARA DOS JUGADORES *****
30 REM ***** AMSTRAD  USER 1986 *****
40 REM
50 DEFINT a-z
60 MODE 0
70 GOSUB 1050
80 GOSUB 1470
90 GOSUB 280
100 GOSUB 1470
110 GOSUB 1360
120 '
130 '
140 REM comienzo
150 IF fin THEN GOTO 100
160 GOSUB 240
170  CALL &BD19:IF p1dir THEN GOSUB 580 ELSE CALL &BD19:CALL &BD19
180  CALL &BD19:IF p2dir THEN GOSUB 640 ELSE CALL &BD19:CALL &BD19
190 IF p1sa=-1 THEN GOSUB 700
200 IF p2sa=-1 THEN GOSUB 760
210 GOTO 140
220 '
230 '
240 IF j THEN  380 ELSE  480
250 '
260 '
270 '
280 CLS:PEN 6
290 PRINT:PRINT" CONTROLES DEL JUEGO"
300 PRINT:PRINT:PRINT:PRINT"TECLADO (T)":PRINT:PRINT TAB (11)," O JOYSTICK (J)"
310 IF NOT J THEN LOCATE 17,10:PRINT" ":LOCATE 13,7:PRINT"*"ELSE LOCATE 13,7:PRI
NT" ":LOCATE 17,10:PRINT"*"
320 IF NOT(INKEY(51)) THEN j=0
330 IF NOT(INKEY(45)) THEN j=-1
340 IF NOT(INKEY(18)) THEN RETURN ELSE 310
350 '
360 '
370 '
380 p1=JOY(0):p2=JOY(1)
390 p1dir=(p1 AND 1)*-1+(p1 AND 2)*0.5
400 p2dir=(p2 AND 1)*-1+(p2 AND 2)*0.5
410 IF P1 AND 16 THEN p1sa=p1sa-1:IF p1sa=-1 THEN AFTER 15 GOSUB 820
420 IF P2 AND 16 THEN p2sa=p2sa-1:IF p2sa=-1 THEN AFTER 15 GOSUB 820
430 IF p1sa THEN p1dir=0
440 IF p2sa THEN p2dir=0
450 RETURN
460 '
470 '
480 p2dir=((INKEY(4)=0)*1)+((INKEY(5)=0)*-1)
490 p1dir=((INKEY(69)=0)*1)+((INKEY(71)=0)*-1)
500 IF INKEY(63)=0 THEN p1sa=p1sa-1:IF p1sa=-1 THEN AFTER 15 GOSUB 820
510 IF INKEY(10)=0 THEN p2sa=p2sa-1:IF p2sa=-1 THEN AFTER 15 GOSUB 820
520 IF p1sa THEN p1dir=0
530 IF p2sa THEN p2dir=0
540 RETURN
550 '
560 '
570 '
580 pt=p1wp+p1dir:IF pt>25 OR pt<6 THEN RETURN ELSE p1wp=pt
590 p1dir=0
```

PROGRAMAS

```
600 PEN 1:LOCATE 3,p1wp:CLS #3:PRINT CHR$(209);:RETURN
610 '
620 '
630 '
640 pt=p2wp+p2dir:IF pt>25 OR pt<6 THEN RETURN ELSE p2wp=pt
650 p2dir=0
660 PEN 2:LOCATE 18,p2wp:CLS #5:PRINT CHR$(211);:RETURN
670 '
680 '
690 '
700 PAPER #4,4:WINDOW #4,4,17,p1wp,p1wp:CLS#4:CALL &BD19:CALL &BD19
710 PAPER #4,0:CLS#4
720 GOTO 580
730 '
740 '
750 '
760 PAPER #6,5:WINDOW #6,4,17,p2wp,p2wp:CLS#6:CALL &BD19:CALL &BD19
770 PAPER #6,0:CLS#6
780 GOTO 640
790 '
800 '
810 '
820 pwpe=(p1wp=p2wp):IF p1sa AND NOT(p2sa) AND pwpe THEN p1sc=p1sc+1:SOUND 132,1
20,10,0,1,0:PRINT#1,a$(p1sc);:IF p1sc=9 THEN 920
830 IF p2sa AND NOT(p1sa) AND pwpe THEN p2sc=p2sc+1:SOUND 132,100,10,0,1,0:PRINT
#2,a$(p2sc);:IF p2sc=9 THEN 920
840 IF p1sa THEN SOUND 132,40,70,0,1,1
850 IF p2sa THEN SOUND 132,56,70,0,1,1
860 p1sa=0
870 p2sa=0
880 RETURN
890 '
900 '
910 '
920 PEN 6
930 LOCATE 5,10:PRINT"FINAL PARTIDA"
940 IF p1sc=9 THEN INK 1,2,20:INK 2,0 ELSE INK 2,6,17:INK 1,0
950 SOUND 129,1000,0,12,3:SOUND 130,900,0,12,3
960 WHILE INKEY$<>"":WEND
970 t!=TIME:WHILE t!+2000>TIME:WEND
980 WHILE INKEY$="":WEND
990 CLS
1000 fin=-1
1010 RETURN
1020 '
1030 '
1040 '
1050 a$(0)="111101101101111"
1060 a$(1)="011001001001001"
1070 a$(2)="111001111100111"
1080 a$(3)="111001111001111"
1090 a$(4)="100100101111001"
1100 a$(5)="111100111001111"
1110 a$(6)="111100111101111"
1120 a$(7)="111001001010010"
1130 a$(8)="111101111101111"
1140 a$(9)="111101111001001"
1150 FOR n=0 TO 9
1160   ho=LEN(a$(n))
1170   FOR n2=1 TO ho
```

```
1180    IF MID$(a$(n),n2,1)="1"THEN MID$(a$(n),n2,1)=CHR$(143)ELSE MID$(a$(n),n2
,1)=CHR$(32)
1190 NEXT n2,n
1200 '
1210 '
1220 b$="LUCHA ELECTRICA"
1230 c$=CHR$(32)+CHR$(164)+" Alexander Martin"
1240 ENV 1,=9,2000:ENT -1,6,3,1
1250 ENV 2,127,0,0,127,0,0,127,0,0,127,0,0,127,0,0
1260 ENV 3,=9,9000
1270 '
1280 '
1290 '
1300 BORDER 0
1310 PEN #4,1:PEN #6,2:PEN #1,1:PEN #2,2:PAPER #1,3:PAPER #2,3:PEN #0,6
1320 RETURN
1330 '
1340 '
1350 '
1360 INK 0,12:INK 1,2:INK 2,6:INK 3,13:INK 4,20:INK 5,17:INK 6,20
1370 WINDOW #3,3,3,6,25:WINDOW #5,18,18,6,25
1380 WINDOW #1,3,5,1,5:WINDOW #2,16,18,1,5:WINDOW #7,1,20,1,5:PAPER #7,3
1390 CLS:CLS#7:PRINT#1,a$(0);:PRINT#2,a$(0);:p1sc=0:p2sc=0:p1wp=5:p2wp=24:p1dir=
1:p2dir=1
1400 GOSUB 580:GOSUB 640
1410 SOUND 1,1000,0,12,2:SOUND 2,900,0,12,2
1420 p1sa=0:p2sa=0:fin=0
1430 RETURN
1440 '
1450 '
1460 '
1470 CLS
1480 PEN 7
1490 FOR n=1 TO LEN(b$)
1500  LOCATE 2+n,10
1510  FOR n2=LEN(b$) TO n STEP-1
1520   PRINT MID$(b$,n2,1)
1530   LOCATE 2+n,10
1540   SOUND 135,20*n2,5,12,2,1
1550 NEXT n2,n
1560 SOUND 135,100,0,13,3,1,20
1570 PEN 6:PRINT:PRINT:PRINT:PRINT c$
1580 t!=TIME:WHILE t!+2500>TIME:WEND
1590 RETURN
```

AMSTRAD CPC - 464

AMSTRAD



ORDENADOR

Esta es la familia de ordenadores personales AMSTRAD. Una familia completa en la que se incluye desde el equipo básico de introducción a la informática hasta el orientado a aplicaciones profesionales. Todo con la filosofía de diseño AMSTRAD que ofrece ordenadores compactos, listos para funcionar sin cableados engorrosos ni necesidad de adquirir periféricos **-con un solo cable a la red-** e incluyendo paquetes de programas de obsequio.

Todos con una tecnología contrastada y fiable basada en el microprocesador Z 80 A, en el Sistema Operativo CP/M - el más extendido para ordenadores de 8 bits- y en una electrónica depurada y con un riguroso control de calidad.

Todos con una extensa biblioteca de programas que se incrementa día a día con títulos para todos los gustos y necesidades.

Todos con una asistencia técnica rápida y eficaz que AMSTRAD ESPAÑA garantiza exclusivamente a los equipos adquiridos a través de su Red Oficial de Distribuidores y acompañados de la Tarjeta de Garantía de AMSTRAD ESPAÑA.

Todos a unos precios increíbles que no admiten comparación con los de cualquier otro ordenador personal de sus características y prestaciones.

AMSTRAD CPC 464.

• Microprocesador Z 80 A• 64K RAM• 32K ROM• Teclado profesional con 32 teclas programables. Sonido estéreo con 3 canales y 8 octavas. Resolución de hasta 640 x 200 puntos. Texto de 20, 40 y 80 columnas. 27 colores. Conectores multiuso, Centronics, joystick etc... Magnetófono incorporado.

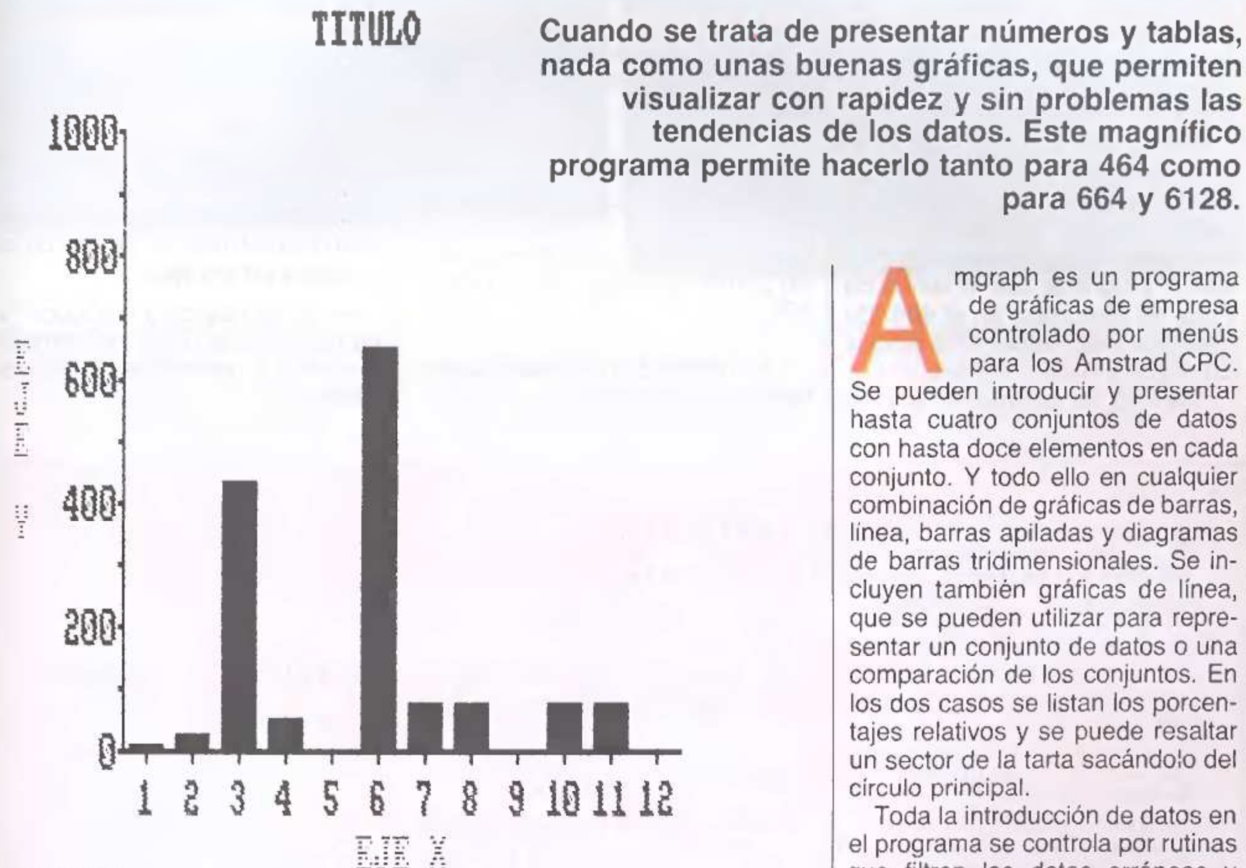
TODO POR: 59.000 pts. (monitor verde)
90.000 pts. (monitor color)

EL SUMINISTRO INCLUYE:

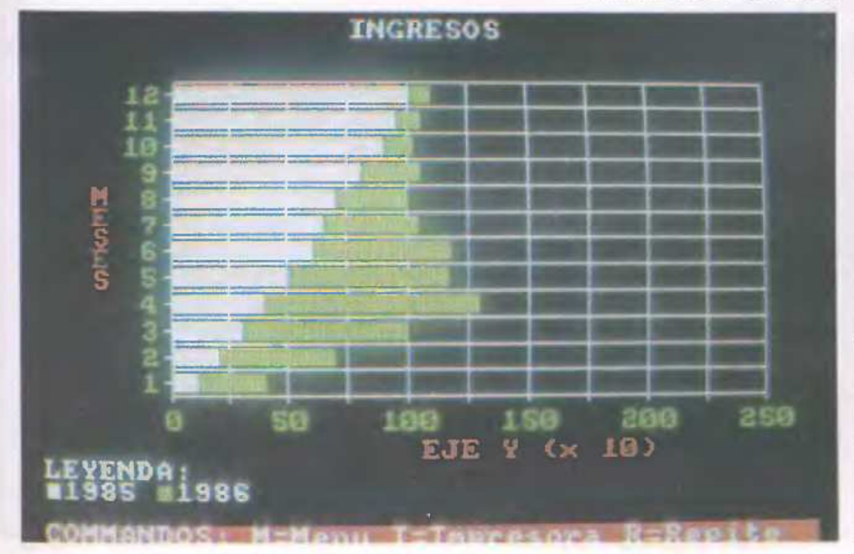
- LIBRO "Guía de Referencia del Programador"
- Manual en castellano
- **8 programas de obsequio en cassette** ("Animal, Vegetal y Mineral", "Amsdraw", "Plaga Galáctica", "Fruit Machine", "Admiral Graph Spee", "Amsword", "El Laberinto del Sultán", "OH. Mummy")



Cuando se trata de presentar números y tablas, nada como unas buenas gráficas, que permiten visualizar con rapidez y sin problemas las tendencias de los datos. Este magnífico programa permite hacerlo tanto para 464 como para 664 y 6128.



BARRAS HORIZONTALES



Amgraph es un programa de gráficas de empresa controlado por menús para los Amstrad CPC. Se pueden introducir y presentar hasta cuatro conjuntos de datos con hasta doce elementos en cada conjunto. Y todo ello en cualquier combinación de gráficas de barras, línea, barras apiladas y diagramas de barras tridimensionales. Se incluyen también gráficas de línea, que se pueden utilizar para representar un conjunto de datos o una comparación de los conjuntos. En los dos casos se listan los porcentajes relativos y se puede resaltar un sector de la tarta sacándolo del círculo principal.

Toda la introducción de datos en el programa se controla por rutinas que filtran los datos erróneos y mantienen una presentación correcta de la pantalla. La introducción de datos se hace en una pantalla diferente de la que se utilizará para dibujar, para que el usuario no pierda la pista a los datos introducidos. Se dispone también de opciones de carga y salvación de datos, así como de modificación.

El formato ganeral admite que se introduzcan etiquetas para los datos y una opción que numera automáticamente según los meses del año. Por ejemplo, una entrada de 5 daría como resultado los meses de mayo a abril.

Las gráficas se escalan automáticamente, aunque se puede poner la línea base para comparar o forzar un rango de escala determinado. Esto resulta muy útil para comparaciones con un nivel dado.

El programa incluye aproximadamente 700 bytes de código máquina que dibuja las barras y proporciona un volcado de pantalla con sombras de gris para impresoras tipo *Epson*. La rutina funciona con la impresora Indescomp.

El programa se debe salvar a cassette o disco antes de ejecutarlo, ya que un fallo en la introducción del código máquina puede dejarlo *colgado*, causando la pérdida de nuestros datos. El programa incluye sumas de control para minimizar este problema.

El programa es compatible CPC 464 y CPC 664 y 6128. Los úsuarios de estos últimos, sin embargo, deben cambiar la línea 40 como se indica. El problema surge porque el programa usa la sentencia FILL, no disponible en el 464. Su uso hace más vistosos los dibujos en el 664 y 6128.

Cuando el programa ejecute correctamente y se corrijan los errores de introducción que aparecerán inevitablemente, la espera mientras se introduce en memoria el código máquina se puede acortar haciendo los siguiente:

Salga del programa mediante BREAK. Cambie la línea 60 a:

```
MEMORY 34999:
LOAD«GRAPHCM, BIN»
```

Borre a continuación las líneas 4000-4160 y salve la nueva versión con:

```
SAVE «AMSGRAPH»
SAVE «GRAPHCM»,B,35059,691
```

Para continuar con el programa sin perder los datos teclee GOTO 190.

Las variables principales usadas por el programa son:

— n: número de conjuntos datos.

— a$(): vector de nombres de los datos.

— d(): tabla de datos.

— e(): vector con las longitudes de los conjuntos de datos.

— month, month$.
Cadena con el mes y el año iniciales:

— msc: línea base para las gráficas

— t$, s$, x$, y$.
Etiquetas de las gráficas

— sc, s, s5:
Factor de escala, potencia de diez (orden de magnitud de los datos) y si es menor que 2,5, 5 ó 10 usada para limitar el tamaño de los números en los ejes.

— k: Se usa para introducir variables para la rutina de introducción d$, d1. cadena de IMPUT y su longitud.

```
10 REM**********************************
20 REM*   AMGRAPH por John Palmer      *
30 REM**********************************
40 cpc464=-1:REM PONER A O PARA UTILIZAR EN UN 664 o 6128
50 SYMBOL AFTER 229:OPENOUT "dummy":MEMORY HIMEM-1:CLOSEOUT
60 MEMORY 34999:GOSUB 4000:RESTORE
70 MODE 1:INK 0,0:INK 1,26:INK 2,18:INK 3,6:BORDER 0:WINDOW#1,1,40,25,25:PAPER#1
,3:WINDOW#2,1,40,24,25:PAPER#2,3:WINDOW#3,1,40,24,25
:DEG
80 DIM opt$(9),b$(5),a$(4),d(12,4),e(4):e(0)=12:FOR f=0 TO 9:READ opt$(f):NEXT:F
OR f=0 TO 4:READ b$(f):NEXT:t$="TITULO":x$="EJE X":y
$="EJE Y"
90 DATA AMGRAPH,Entrada de datos iniciales,anadir/modificar/visualizar,Formato g
eneral,Grafico de barra (vert),Grafico de linea,Graf
ico de tarta,Grafico de barra (horiz),Grafico de barra 3D,Cargar/Salvar datos
100 DATA Introduzca el nombre de los datos,Introduzca datos S=salir,Introduzca o
pcion requerida,Introduzca numero,Introduzca nuevos
detalles
110 FOR f=229 TO 255:IF f=233 THEN 130
120 READ x1,x2,x3,x4,x5,x6:SYMBOL f,0,x1,x2,x3,x4,x5,x6,0
130 NEXT:FOR f=234 TO 255:year$=year$+CHR$(f):IF f=241 THEN year$=year$+CHR$(238
)
140 IF f=244 THEN year$=year$+CHR$(243)
150 NEXT
160 DATA 126,0,126,0,126,0,0,0,146,0,0,0,0,0,254,0,0,0,42,42,42,42,42,42,249,18,
18,19,146,98,145,89,93,215,83,81,251,130,243,130,130
,131,222,17,222,17,17,222,137,218,250,171,138,138,158,81,81,222,82,81
170 DATA 115,138,138,251,138,138,158,81,81,158,18,17,145,81,74,196,68,68,250,18,
18,18,146,97,81,89,93,87,83,145,80,80,80,80,80,159,1
14,138,139,250,133,137,73,81,80,93,81,142,123,130,115,10,138,115,222,17,209,30,1
6,208
180 DATA 113,138,138,135,138,113,223,4,4,4,4,196,137,202,234,186,154,137,145,81,
74,74,68,132,227,146,139,138,146,227,206,17,208,16,1
7,206
190 REM**********************************
200 REM*            Menu                *
210 REM**********************************
220 ms=0:GOSUB 2190
```

AMGRAPH

1) Entrada de datos iniciales
2) anadir/modificar/visusalizar
3) Formato general
4) Grafico de barra (vert)
5) Grafico de linea
6) Grafico de tarta
7) Grafico de barra (horiz)
8) Grafico de barra 3D
9) Cargar/Salvar datos

Selecciona numero de opcion

MENU

CARGAR/SALVAR




```
230 FOR f=1 TO 9:LOCATE 7,f*2+3:PRINT USING "#";f;:PRINT") ";opt$(f):NEXT
240 CLS#1:PRINT#1,TAB(9):"Selecciona numero de opcion"
250 i$=INKEY$:IF i$="" THEN 250 ELSE ms=ASC(i$)-48:IF ms<1 OR ms>9 THEN 250
260 IF n=0 AND ms<>3 AND ms>1 AND ms<9 THEN CLS#1:PRINT#1,TAB(13)"No hay datos "
:FOR b=1 TO 500:NEXT:GOTO 240
270 ON ms GOTO 280,400,650,1000,1060,1180,1470,1590,1710
280 REM*******************************
290 REM*        Datos iniciales      *
300 REM*******************************
310 CLS:IF n=0 THEN 350
320 LOCATE 16,12:PRINT"CUIDADO":LOCATE 4.14:PRINT"SE HAN INTRODUCIDO DATOS ANTER
IORMENTE":LOCATE 6,16:PRINT" Desea continuar  (S/N)"
330 i$=UPPER$(INKEY$):IF i$="N" THEN 190 ELSE IF i$<>"S" THEN 330
340 ERASE a$,d,e:DIM a$(4),d(12,4),e(4):e(0)=12
350 g=0:GOSUB 2190:FOR f=1 TO 12:LOCATE 1,8+f:PRINT USING "##";f;:PRINT")":NEXT
360 g=g+1:GOSUB 1850:IF g=4 THEN 390
370 CLS#1:PRINT#1,"Desea introducir algun dato mas (S/N)"
380 i$=UPPER$(INKEY$):IF i$="S" THEN 360 ELSE IF i$<>"N" THEN 380
390 n=g:GOTO 190
400 REM*******************************
410 REM*        visualizar datos     *
420 REM*******************************
430 GOSUB 2190:GOSUB 2590:FOR f=1 TO 12:LOCATE 1,8+f:PRINT USING "##";f;:IF mont
h=0 THEN PRINT ")" ELSE PEN 2:PRINT MID$(month$,f*2-
1,2):PEN 1
440 NEXT:FOR g=1 TO n:LOCATE g*9,6:PRINT USING"#";g;:PRINT")":LOCATE g*9+4-LEN(a
$(g)),7:PRINT a$(g)
450 FOR f=1 TO e(g):c$=STR$(d(f,g)):LOCATE g*9+4-LEN(c$),8+f:PRINT c$:NEXT:NEXT
460 CLS#2:PRINT#2,"COMMANDOS: M=Menu 1=Canbiar  2=anadir            3=cambiar nomb
re 4=impresora"
470 i$=UPPER$(INKEY$):IF i$="" THEN 470 ELSE IF i$="M" THEN 190 ELSE op=ASC(i$)-
48:IF op<1 OR op>4 THEN 470
480 IF op=4 THEN CLS#3:GOSUB 2870:GOTO 460
490 ON op GOSUB 500,570,610:GOTO 460
500 CLS#3:g=1:a=16:IF n>1 THEN GOSUB 630:a=0
510 LOCATE 17-a,22:PRINT"Fila num":c$=b$(3)
520 col=30-a:k=7:GOSUB 1910:f=VAL(d$):IF f<1 OR f>12 THEN 520
530 LOCATE 1,22:PRINT SPACE$(32)
540 k=0:GOSUB 1910:IF i=endat OR i=endat+32 THEN d$=STR$(d(f,g)):dl=LEN(d$):GOTO
 2170
550 d(f,g)=VAL(d$):IF f>e(g) THEN e(g)=f
560 RETURN
570 IF n<4 THEN n=n+1:g=n:GOTO 600
580 CLS#3:GOSUB 630:CLS#2:PRINT#2,TAB(9)"La calunna "+STR$(g)+" Ya contiene dato
s.";TAB(11)"Desea continuar  (S/N)"
590 i$=UPPER$(INKEY$):IF i$="N" THEN 460 ELSE IF i$<>"S" THEN 590
```

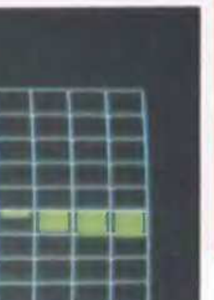
```
600 CLS#3:FOR f=1 TO 12:d(f,g)=0:LOCATE g*9-3,f+8:PRINT SPACE$(7):NEXT:GOSUB 185
0:RETURN
610 CLS#3:g=1:IF n>1 THEN GOSUB 630
620 k=1:GOSUB 1910:a$(g)=d$:RETURN
630 LOCATE 1,22:PRINT"Columna num":k=3:GOSUB 1910:g=VAL(d$):IF op<>1 THEN LOCATE
 1,22:PRINT SPACE$(16)
640 RETURN
650 REM*******************************
660 REM*          Formato            *
670 REM*******************************
680 GOSUB 2190:LOCATE 1,5:PRINT"1) Titulo: ":PRINT TAB(4);t$
690 PRINT:PRINT:PRINT"2) Sub-titulo: ":PRINT TAB(4);s$
700 PRINT:PRINT:PRINT"3) Etiqueta  X:   ";x$
710 PRINT:PRINT:PRINT"4) Etiqueta  Y:   ";y$
720 PRINT:PRINT:PRINT"5) Nivel de graficos:       ";msc
730 PRINT:PRINT:PRINT"6) Opcion ano:   ";
740 IF month=0 THEN PRINT"No"; ELSE PRINT"Si"
750 LOCATE 22,22:PRINT"Primer mes ?:";month
760 CLS#1:PRINT#1,"  Pulsa numero a cambiar   M=MENU"
770 i$=UPPER$(INKEY$):IF i$="" THEN 770 ELSE IF i$="M" THEN 190 ELSE op=ASC(i$)-
48:IF op<1 OR op>6 THEN 770
780 col=4:length=36:IF op=1 THEN lin=6:GOSUB 890:t$=d$
790 IF op=2 THEN lin=10:GOSUB 890:s$=d$
800 col=18:length=12:IF op=3 THEN lin=13:GOSUB 890:x$=d$
810 IF op=4 THEN lin=16:GOSUB 890:y$=d$
820 IF op=5 THEN k=6:GOSUB 1910:msc=VAL(d$)
830 IF op<>6 THEN 760
840 CLS#1:PRINT#1,TAB(6)"Opcion anual  ? (S/N)"
850 i$=UPPER$(INKEY$):IF i$="N" THEN month=0:LOCATE 16,22:PRINT"NO":LOCATE 35,22
:PRINT"0 ":GOTO 760
860 IF i$<>"S" THEN 850 ELSE LOCATE 16,22:PRINT"SI "
870 c$="    Introduzca el mes de comienzo":col=37:k=7:GOSUB 1910:month=VAL(d$):IF
 month<1 OR month>12 THEN 870
880 month$=RIGHT$(year$,(13-month)*2)+LEFT$(year$,(month-1)*2):GOTO 760
890 PAPER 2:LOCATE col,lin:PEN 3:PRINT CHR$(143);:PEN 0:PRINT SPACE$(length):d$=
"":IF ms=3 THEN c$=b$(4) ELSE c$=b$(0)
900 CLS#1:PRINT#1,TAB(21-LEN(c$)/2)c$
910 i$=INKEY$:IF i$="" THEN 910 ELSE i=ASC(i$)
920 IF i=127 THEN IF d$<>"" THEN LOCATE col+LEN(d$),lin:PRINT" ":d$=LEFT$(d$,LEN
(d$)-1):GOTO 970 ELSE 910
930 IF i=13 THEN IF ms=9 AND d$="" THEN 980 ELSE 990
940 IF LEN(d$)=length THEN 980
950 IF i<32 OR i>126 THEN 980
960 d$=d$+i$
970 LOCATE col,lin:PRINT d$;:PEN 3:PRINT CHR$(143):PEN 0:GOTO 910
980 CLS#1:PRINT#1,TAB(10)"** Entrada invalida **":SOUND 1,75:FOR b=1 TO 500:NEXT
:GOTO 900
990 PAPER 0:PEN 1:LOCATE col,lin:PRINT d$+SPACE$(1+length-LEN(d$)):RETURN
1000 REM*******************************
1010 REM*     GRAFICO DE BARRAS       *
1020 REM*******************************
1030 GOSUB 2190:GOSUB 2230
1040 IF dl=1 THEN wid=30:disp=0 ELSE IF dl=2 THEN wid=14:disp=15 ELSE IF dl=3 TH
EN wid=12:disp=8 ELSE wid=10:disp=7
1050 POKE &88EE,wid:POKE &88EF,disp:CALL &88F3:GOTO 2730
1060 REM*******************************
1070 REM*     GRAFICO DE LINEAS       *
1080 REM*******************************
1090 GOSUB 2190:GOSUB 2230
1100 FOR a=1 TO dl:g=VAL(MID$(d$,a,1))
1110 IF NOT cpc464 AND g=4 THEN MASK 136,0:GRAPHICS PEN ,1
1120 x1=132:y1=112+(120 AND m=0.5)+(d(1,g)-msc)*sc:PLOT 1000,1000,g-(2 AND g=4):
FOR f=2 TO e(g)
1130 fig%=(d(f,g)-d(f-1,g))*sc:IF NOT (cpc464 AND g=4) THEN MOVE x1,y1:DRAW x1+4
0,y1+fig%
1140 IF cpc464 AND g=4 THEN hyp=SQR(1600+fig%*fig%):x=40/hyp:y=fig%/hyp:FOR b=0
```



BARRAS

AMGRAPH

BARRAS



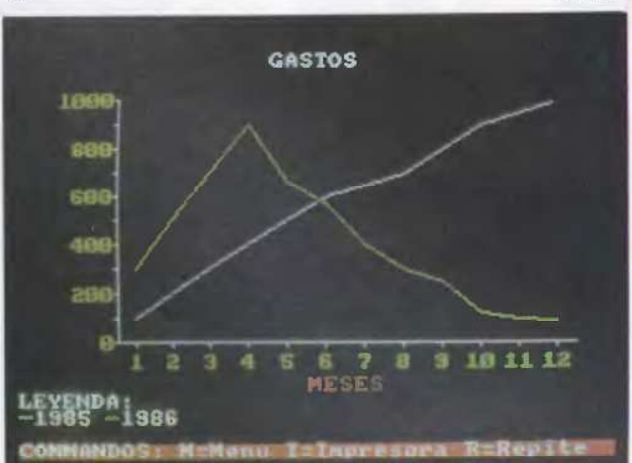

```
TO hyp STEP 8:PLOT x1+b*x,y1+b*y:NEXT
1150 x1=x1+40:y1=y1+fig%:NEXT
1160 IF NOT cpc464 AND g=4 THEN MASK 255,1
1170 NEXT:PLOT 1000,1000,1:GOTO 2730
1180 REM*******************************
1190 REM*      Graficos de tarta      *
1200 REM*******************************
1210 GOSUB 2190:h=0:cross=0:IF n=1 THEN g=n:fin=e(g):IF e(g)<2 THEN 1320 ELSE 12
90
1220 LOCATE 1,5:PRINT"Por filas o por colunnas (F/C)"
1230 i$=UPPER$(INKEY$):IF i$<>"F" AND i$<>"C" THEN 1230
1240 LOCATE 32,5:PRINT i$:IF i$="C" THEN 1270
1250 cross=1:fin=n:LOCATE 1,12:PRINT"Fila num:":k=5:GOSUB 1910
1260 f=VAL(d$):IF f>MAX(e(1),e(2),e(3),e(4)) THEN 1250 ELSE 1290
1270 GOSUB 2230:LOCATE 1,16:PRINT"que datos?":k=4:GOSUB 1910:g=VAL(d$):fin=e(g):
IF e(g)<2 THEN 1320
1280 h=0:IF e(g)<2 THEN 1320
1290 CLS#3:LOCATE 1,18:PRINT"Desea algun segmento resaltado  (S/N)"
1300 i$=UPPER$(INKEY$):IF i$="N" THEN 1320 ELSE IF i$<>"S" THEN 1300
1310 LOCATE 31,19:PRINT i$:LOCATE 1,22:PRINT"Segmento num ":c$="Introduzca numer
o a resaltar":col=18:k=7:GOSUB 1910:h=VAL(d$)
1320 tot=0:FOR a=1 TO fin:IF cross=0 THEN f=a ELSE g=a
1330 tot=tot+ABS(d(f,g)):NEXT
1340 GOSUB 2580:LOCATE 30-cross*6,4:PRINT"Porcentajes":LOCATE 1,4:IF cross=0 THE
N PRINT a$(g) ELSE  IF month=0 THEN PRINT"Fila:";f E
LSE PRINT"MES:";MID$(month$,f*2-1,2)
1350 numb$="#":var=0:FOR a=1 TO fin:ORIGIN 200,184:IF cross=0 THEN f=a ELSE g=a
1360 seg=360*ABS(d(f,g))/tot:IF a=h THEN ORIGIN 200+20*SIN(var+seg/2),184+20*COS
(var+seg/2)
1370 b=var:WHILE b<var+seg:DRAW 120*SIN(b),120*COS(b):b=b+6:WEND
1380 var=var+seg:DRAW 120*SIN(var),120*COS(var):DRAW 0,0
1390 MOVE 110*SIN(var-seg/2),110*COS(var-seg/2):IF NOT cpc464 THEN FILL a MOD 4
1400 IF seg<12 AND a<>h THEN PEN 2:LOCATE 29-cross*6,5+a:PRINT"*":LOCATE 29,23:P
RINT"* Sin etiqueta en el grafico":PEN 1:GOTO 1430
1410 MOVE 130*SIN(var-seg/2)+(16+(16 AND (a>9 OR (month<>0 AND cross=0))))*(XPOS
-120)/240,130*COS(var-seg/2)+16*(YPOS+110)/240:TAG:I
F a>9 THEN numb$="##"
1420 IF month<>0 AND cross=0 THEN PRINT MID$(month$,a*2-1,2); ELSE PRINT USING n
umb$;a;
1430 TAGOFF:ORIGIN 200,184:LOCATE 30-cross*6,5+a:IF cross=0 AND month<>0 THEN PR
INT MID$(month$,a*2-1,2); ELSE PRINT USING numb$;a;
1440 PRINT")";:IF cross=1 THEN PRINT a$(g);
1450 PRINT TAB(34) USING "###.##";100*ABS(d(f,g))/tot;:PRINT"%"
1460 NEXT:ORIGIN 0,0:GOTO 2810
1470 REM*********************************
1480 REM* GRAFICO DE BARRAS HORIZONTALES  *
```

```
1490 REM**********************************
1500 GOSUB 2190:GOSUB 2230
1510 PLOT 112,352:DRAWR 0,-240:DRAWR 480,0
1520 IF grid=1 THEN FOR f=1 TO 12:MOVE 112,f*20+112:DRAWR 480,0:NEXT
1530 TAG:FOR f=1 TO 12:PLOT 106,f*20+102,1:DRAWR 4,0:PLOT 1000,1000,2:MOVE 70,f*
20+108:IF month=0 THEN PRINT USING "##";f; ELSE PRIN
T MID$(month$,f*2-1,2);
1540 NEXT:FOR f=0 TO 10:PLOT 112+f*48,108,1:DRAWR 0,4+(240 AND grid=1):NEXT
1550 PLOT 1000,1000,2:FOR f=0 TO 10 STEP 2:MOVE 106+f*48,100:GOSUB 2910:NEXT
1560 PLOT 1000,1000,3:MOVE 354-16*LEN(y$)/2,78:PRINT y$;:IF s>2 THEN PRINT" (x";
STR$(10^(s-2));")";
1570 MOVE 60,246+16*LEN(x$)/2:FOR f=1 TO LEN(x$):MOVER -16,-16:PRINT MID$(x$,f,1
);:NEXT
1580 TAGOFF:CALL &8965:GOTO 2730
1590 REM*******************************
1600 REM*          GRAFICOS 3D        *
1610 REM*******************************
1620 GOSUB 2190:GOSUB 2230
1630 FOR b=0 TO dl:MOVE 160+b*20,276+b*20:DRAWR 0,-100:DRAWR 384,-144:NEXT
1640 FOR b=176 TO 276 STEP 20:MOVE 156,b:DRAWR 4,0:DRAWR dl*20,dl*20:DRAWR 384,-
144:NEXT
1650 FOR b=1 TO 12:MOVE 160+b*32,176-b*12:DRAWR dl*20,dl*20:DRAWR 0,100:NEXT
1660 TAG:PLOT 1000,1000,2:FOR f=0 TO 10 STEP 2:MOVE 156,182+f*10:GOSUB 2910:NEXT
1670 GOSUB 2960:PLOT 1000,1000,2:FOR b=0 TO 11:MOVE 152+b*32,166-b*12
1680 IF month=0 THEN MOVER -4,0:PRINT USING "##";b+1; ELSE PRINT MID$(month$,b*2
+1,2);
1690 NEXT:PLOT 1000,1000,3:MOVE 278-20*LEN(x$)/2,90+8*LEN(x$)/2:FOR b=1 TO LEN(x
$):MOVER 4,-8:PRINT MID$(x$,b,1);:NEXT:TAGOFF
1700 CALL &89DD:GOTO 2730
1710 REM*******************************
1720 REM*   CARGAR/SALVAR DATOS       *
1730 REM*******************************
1740 GOSUB 2190:LOCATE 1,5:PRINT"CARGAR/SALVAR o RETORNAR? (C/S/R) ";
1750 s=0:i$=UPPER$(INKEY$):IF i$="R" THEN 190 ELSE IF i$="S" THEN s=1 ELSE IF i$
<>"C" THEN 1750
1760 IF s=1 AND n=0 THEN CLS#1:PRINT#1,TAB(12)"No hay datos":FOR b=1 TO 1000:NEX
T:CLS#3:GOTO 1750
1770 PRINT i$:LOCATE 1,12:PRINT"NOMBRE?: ":length=8:lin=12:col=16:GOSUB 890
1780 PRINT:PRINT:IF s=1 THEN 1820
1790 OPENIN d$:INPUT#9,n,msc,t$,s$,x$,y$,month:IF month<>0 THEN INPUT#9,month$
1800 FOR g=1 TO n:INPUT#9,e(g),a$(g):FOR f=1 TO 12:INPUT#9,d(f,g):NEXT:NEXT
1810 CLOSEIN:GOTO 190
1820 OPENOUT d$:WRITE#9,n,msc,t$,s$,x$,y$,month:IF month<>0 THEN WRITE#9,month$
1830 FOR g=1 TO n:WRITE#9,e(g),a$(g):FOR f=1 TO 12:WRITE#9,d(f,g):NEXT:NEXT
1840 CLOSEOUT:GOTO 190
1850 REM*******************************
1860 REM*      ENTRADA DE DATOS       *
1870 REM*******************************
1880 LOCATE g*9,6::PRINT USING"#";g;:PRINT")":k=1:GOSUB 1910:a$(g)=d$
1890 FOR f=1 TO 12:k=0:GOSUB 1910:IF i=endat OR i=endat+32 THEN 1900 ELSE d(f,g)
=VAL(d$):NEXT
1900 e(g)=f-1:RETURN
1910 REM*******************************
1920 REM* RUTINA PRINCIPAL DE TECLADO *
1930 REM*******************************
1940 IF k=0 THEN lin=f+8:length=7:maxim=57:minim=45:c$=b$(1):endat=69:excep=47 E
LSE endat=256:excep=0
1950 IF k=1 THEN lin=7:length=8:maxim=127:minim=31:c$=b$(0)
1960 IF k=2 THEN lin=18:col=24:length=n:c$=b$(2)
1970 IF k=3 THEN lin=22:length=1
1980 IF k=4 THEN lin=16:length=1
1990 IF k=5 THEN lin=12:length=2
2000 IF k=6 THEN lin=19:col=34:length=6
2010 IF k=7 THEN lin=22:length=2
2020 IF k<2 THEN col=g*9+4 ELSE IF k>2 AND k<6 THEN col=16
2030 IF k>4 THEN maxim=57:minim=48 ELSE IF k>1 THEN maxim=n+48:minim=49
```

AMGRAPH



LINEA



LINEA

```
2040 IF k>2 AND k<7 THEN c$=b$(3)
2050 PAPER 2:PEN 0:LOCATE col-length,lin:PRINT SPACE$(length):d$="":dl=0
2060 CLS#1:PRINT#1,TAB(21-LEN(c$)/2)c$
2070 i$=INKEY$:IF i$="" THEN 2070 ELSE i=ASC(i$)
2080 IF i=127 THEN IF dl<>0 THEN LOCATE col-dl,lin:PRINT " ":d$=LEFT$(d$,dl-1):GO
TO 2160 ELSE 2070
2090 IF i=13 AND dl<>0 THEN 2170
2100 IF dl=length THEN 2180
2110 IF i=endat OR i=endat+32 AND d$="" THEN 2170
2120 IF i=46 AND k=0 THEN IF INSTR(d$,i$)<>0 THEN 2180
2130 IF dl>1 AND i=45 AND k=0 THEN 2180
2140 IF i<minim OR i>maxim OR i=excep THEN 2180
2150 d$=d$+i$
2160 dl=LEN(d$):LOCATE col-dl,lin:PRINT d$:GOTO 2070
2170 PAPER 0:PEN 1:LOCATE col-length,lin:PRINT SPACE$(length-dl)+d$:RETURN
2180 CLS#1:PRINT#1,TAB(10)"** Entrada invalida **":SOUND 1,75:FOR b=1 TO 500:NEX
T:GOTO 2060
2190 REM*******************************
2200 REM*      IMPRIMIR CABECERA      *
2210 REM*******************************
 2220 CLS:PAPER 2:PEN 0:LOCATE (21-LEN(opt$(ms))/2),1:PRINT opt$(ms):PAPER 0:PEN
1:PRINT:RETURN
2230 REM*******************************
2240 REM*         Que dato ?          *
2250 REM*******************************
2260 IF n=1 THEN d$="1":dl=1:IF ms=8 THEN 2390 ELSE 2330
2270 PRINT:PRINT:PRINT"Que datos?":PRINT
2280 FOR f=1 TO n:PRINT USING"#";f;:PRINT") ";a$(f):NEXT:IF ms=6 THEN RETURN
2290 LOCATE 1,18:PRINT"Datos a representar:":k=2:GOSUB 1910:CLS:IF ms=8 THEN 236
0
2300 REM*******************************
2310 REM*    Tendido de lineas?       *
2320 REM*******************************
2330 LOCATE 4,12:PRINT"Desea tendido de lineas  (S/N)"
2340 i$=UPPER$(INKEY$):IF i$="S" THEN grid=1 ELSE IF i$="N" THEN grid=0 ELSE 234
0
2350 CLS
2360 REM*******************************
2370 REM*          ESCALA             *
2380 REM*******************************
2390 LOCATE 14,10:PRINT"ESPERE UN MOMENTO"
2400 IF ms=7 THEN FOR f=1 TO 12:d(f,0)=0:FOR a=1 TO dl:g=VAL(MID$(d$,a,1)):d(f,0
)=d(f,0)+d(f,g):NEXT:NEXT:dl=1
2410 s=0:s5=4:m=1:FOR a=1 TO dl:IF ms=7 THEN g=0 ELSE g=VAL(MID$(d$,a,1))
2420 FOR f=1 TO e(g):k=0:b=ABS(d(f,g)-(msc AND ms<>7))
2430 WHILE b>10:b=b/10:k=k+1:WEND
2440 IF k>s THEN s=k:s5=4
```

```
2450 IF SGN(d(f,g)-msc)=-1 THEN m=0.5
2460 NEXT:FOR f=1 TO e(g):b=ABS(d(f,g)-(msc AND ms<>7))/10^s
2470 IF b>5 THEN s5=1 ELSE IF b>2.5 AND s5<>1 THEN s5=2
2480 NEXT:NEXT:IF s>2 THEN xsc=2 ELSE xsc=s
2490 IF ms=7 THEN m=1:dl=LEN(d$)
2500 sc=24*m*s5/10^s:IF ms=5 THEN 2580 ELSE IF ms=8 THEN sc=sc/2.4
2510 REM********************************
2520 REM*        Pokear  datos         *
2530 REM********************************
2540 POKE 35001,dl:IF ms=4 THEN POKE 35000,114+(120 AND m=0.5) ELSE POKE 35000,1
70+(50 AND m=0.5)
2550 FOR a=dl TO 1 STEP-1:g=VAL(MID$(d$,a,1)):mem=(dl-a)*13+35002:POKE mem,g:mem
=mem+1
2560 FOR f=1 TO e(g):IF ms=7 THEN POKE mem,ABS(d(f,g)*sc) ELSE POKE mem,INT((256
 AND SGN((d(f,g)-msc))=-1)+(d(f,g)-msc)*sc)
2570 mem=mem+1:NEXT:NEXT
2580 CLS
2590 REM********************************
2600 REM*       INPRIMIR TITULOS       *
2610 REM********************************
 2620 PRINT TAB(21-LEN(t$)/2);t$:PRINT TAB(21-LEN(s$)/2);s$:IF ms<4 OR ms>5 THEN
RETURN
2630 REM**********************************
2640 REM*  Dibuja grafico de linea , barras *
2650 REM**********************************
2660 MOVE 112,352:DRAWR 0,-240:MOVE 112,112+(120 AND m=0.5):DRAWR 480,0
2670 FOR f=132 TO 592 STEP 40:MOVE f,108:DRAWR 0,2:IF grid=1 THEN MOVER 20 AND m
s=4,2:DRAWR 0,240
2680 NEXT:FOR f=112 TO 352 STEP 24:MOVE 108,f:DRAWR 4+(480 AND grid=1),0:NEXT
2690 TAG:PLOT 1000,1000,2:FOR f=0 TO 10 STEP 2:MOVE 108,118+f*24:GOSUB 2910:NEXT
2700 GOSUB 2960
2710 PLOT 1000,1000,2:FOR b=0 TO 11:MOVE b*40+116-(8 AND month=0 AND b<9),100:IF
 month=0 THEN PRINT USING "##";b+1; ELSE PRINT MID$(
month$,b*2+1,2);
2720 NEXT:PLOT 1000,1000,3:MOVE 354-16*LEN(x$)/2,79:PRINT x$;:TAGOFF:RETURN
2730 REM********************************
2740 REM*           Leyenda            *
2750 REM********************************
2760 LOCATE 1,22-(3 AND ms=8):PRINT"LEYENDA:"
2770 FOR f=1 TO dl:g=VAL(MID$(d$,f,1)):PEN g-(2 AND g=4)
2780 IF ms<>5 THEN PRINT CHR$(233-(1 AND g=4)-(3 AND g=4 AND ms=7)); ELSE PRINT
CHR$(231-(1 AND g=4));
2790 PEN 1:PRINT a$(g);" ";:IF ms=8 THEN PRINT
2800 NEXT
2810 REM********************************
2820 REM*          Commandos           *
2830 REM********************************
2840 CLS#1:PRINT #1,"COMMANDOS: M=Menu I=Impresora R=Repite"
2850 i$=UPPER$(INKEY$):IF i$="M" THEN 190 ELSE IF i$="R" THEN 270 ELSE IF i$<>"I
" THEN 2850
2860 GOSUB 2870:GOTO 2840
2870 CLS#1:PRINT#1,TAB(14)"Espere un momento"
2880 POKE &8B1C,144-(24 AND ms=2):CALL &8B09:IF PEEK(35000)=0 THEN RETURN
2890 CLS#1:PRINT#1,"NO PREPARADA REINTENTAR/CANCELAR (R/C)"
2900 i$=UPPER$(INKEY$):IF i$="C" THEN RETURN ELSE IF i$="R" THEN 2870 ELSE 2900
2910 REM********************************
2920 REM*       Escala grafica         *
2930 REM********************************
2940 g=(f-(5 AND m=0.5))*10^xsc/(s5*m):IF ms<>7 THEN g=g+INT(msc/10^(s-xsc))
2950 g$=STR$(g):MOVER -16*LEN(g$)/(1+(1 AND ms=7)),0:PRINT g$;:RETURN
2960 x=112+(44 AND ms=8)-16*LEN(g$):y=246+(32 AND ms=8)
2970 IF s>2 THEN f$="x"+STR$(10^(s-2)):MOVE x,y+16*LEN(f$)/2:FOR b=1 TO LEN(f$):
MOVER -16,-16:PRINT MID$(f$,b,1);:NEXT
2980 PLOT 1000,1000,3:MOVE x-(16 AND s>2),y+16*LEN(y$)/2:FOR b=1 TO LEN(y$):MOVE
R -16,-16:PRINT MID$(y$,b,1);:NEXT:RETURN
4000 REM********************************
```



TARTA

TARTA




```
4010 REM*           pokear C/M          *
4020 REM********************************
4030 CLS:PRINT"POKEANDO CODIGO-MAQUINA...... ESPERE"
4040 RESTORE 4080:x=35059:FOR a=0 TO 8:READ a$:tot=0:FOR b=1 TO LEN(a$)/2
4050 c=VAL("&"+MID$(a$,b*2-1,2)):tot=tot+c:POKE x,c:x=x+1:NEXT
4060 READ check:IF tot<>check THEN PRINT"Error en linea ";4080+a*10:STOP
4070 NEXT:POKE 35628,4:POKE 35579,&4C:POKE 35580,64:POKE 35581,1:RETURN
4080 DATA DD21B988DD460026003AB8886FC5E52176003AEF885F1600783D28031918FAEBE1DD23
DD7E0032F088CDDEBB060CC5E5D5DD2306007DFEEADD7E002009
A7F23989ED444FED42D602380D32F1883AEE88,9500
4090 DATA 0F47CD3B8A10FBE10E2809EBE1C110CEC110AA063BAF21B88772310FC3CCDDEBBC93A
B98110D0021BA88193D20FCE5E5DDE1FDE12174000600CC5FDE5
DD23DDE53AB9884711720OC5060DDD2BFD2B10,9708
4100 DATA FAFD7E0032F088CDDEBBE5DD7E00A728283D6F602922F188E1E50607CDE38AE5D5CDD3
8A210000ED5BF188CDEB8AD1E1232310E8CDC6BB1313E1C110BB
0E1409DDE1FDE1C110A3C35689DD21B988DD46,11165
4110 DATA 00C5AFC61410FC4F21780009EB3AB88826006F09C1C5E5D5DD23DD7E0032F088360CC5
DD2306000E20E5EB09EBE10E0CED42DD7E00A72812D5E5F2258A
ED444FED4232F188CD408AE1D1C110D5D1E10E,9772
4120 DATA 14EBED42EBED42C110BBC356893AF088CDDEBB060878FE052804FE0320022B2BCD8B8A
10F0E5D50101042323CD8B8ACDA18A10F6D1E13AF088FE04C81B
1BAF4FCDDEBBCD8B8ACDC6BBCDAD8ACDE38A11,10699
4130 DATA 0800626BCDEB8AC9CDD38ACDE38AE5D52AF188110000CDEB8AD1E11313C93AF088FE04
2005D602CDDEBBE5D5C5CDC6BB06073E0591902804D602200923
2379A72803CDE38A1B1BCDE38A10E6C1D1E1C9,10372
4140 DATA 3AF088FE04C0CB402802D602CDDEBBC9C5D5E5CDEABB1806C5D5E5CDF9BBE1D1C1C91B
401B33121B2A0440010D0A0000030302010200021F58A06057ECD
2BBD380432B888C92310F3219001E51100000E,8339
4150 DATA 0206A0C5E521FA8A0665CD8C8BE1C1C5D5E5CD978B0E02EDB0E1D1D5E50602C52B2BD5
E5CD978B06021A0707B612231310F7E1D1C110E821D988CD8A8B
E1D1C1131310CB0D20BA21FF8ACD8A8BE1A70E,10300
4160 DATA 06ED427CA720A47DFE10209F21F58ACD8A8BAF32B888C906027ECD2BBD30FB7E2310F6
C9CDF0BB21018B11B988706004F09C9,6093
```

AMGRAPH

3-D VOICE CHESS

Ajedrez tridimensional con voz en castellano

Amstrad CPC 464, CPC 664 y CPC 6128

P.V.P.
2.300.- (cinta)
3.300.- (disco)

PUBLISHED
DEEP THOUGHT
SOFTWARE
DISTRIBUTED
cp software

Producido en exclusiva para España por:



Actividades Comerciales Electrónicas, S.A.
Tarragona 110 - Tel. 325 10 58 * 08015 Barcelona. Télex 93133 ACEE E



¿QUE SIGNIFICA TODA ESTA JERGA

Hay una gran definición de la diferencia entre **Hardware** y **Software:** ¡si le puedes dar una patada entonces es **Hardware**! **Hardware** es la electrónica y la carcasa en la cual está montada. **Software** son los **programas** que proyectas sobre el ordenador. Es como un sistema de alta fidelidad —el amplificador, el reproductor de cassettes y los altavoces son **hardware,** mientras que la música que pones es **software.**

El **Hardware** de un micro-ordenador consta de diferentes partes que incluyen un **teclado** para mecanografiar las instrucciones y una pantalla (monitor) para exhibir los resultados. Esta podría ser tu propia televisión o una pantalla especialmente modificada y acoplada al ordenador. Sobre ella podrás representar **caracteres alfanuméricos** (A-Z, 0-9 y puntuación) y **caracteres gráficos** especiales para diagramas y dibujos. Estos caracteres van desde sencillas líneas y cuadrados a complicados símbolos de juegos. Cuanto más alta sea la resolución de tu pantalla, más finas entonces pueden ser dibujadas las líneas y más limpias aparecen las curvas.

Por dentro, el ordenador está compuesto de **circuitos integrados** (IC) o «chips» —pequeños y sólidos componentes electrónicos enfundados en un plástico negro. El más importante de estos circuitos es el llamado «Unidad Central de Proceso» o «CPU», que lleva a cabo todos los cálculos y dirige todo el sistema, trasladando la información a todas las partes. Los dos modelos más comunes de CPU (Unidad Central de Proceso) son el MOS Technology **6502** utilizado en los ordenadores **Commodore, Apple** y **Atari;** y el Zilog **Z-80** utilizado por **Tandy** y **Sinclair** entre otros y por supuesto **Amstrad.** Otros dos CPUs populares son el MOTOROLA 68000 y el INTEL 8088.

El ordenador tiene dos clases de memorias. La Memoria de Acceso aleatorio **(Random Access Memory)** o **RAM** que es una memoria temporal —cualquier cosa almacenada en ella se pierde al desconectar el ordenador— el término técnico es volátil.

La **RAM** se utiliza para almacenar: todos los **comandos** (instrucciones para el ordenador) que mecanografíe, el programa que está habitualmente ejecutándose y los datos (información) a los que se refiere dicho programa. Estos **datos** consisten en «variables»: una variable es sólo una pieza de información y tiene un nombre que la identifica, por ejemplo: PRECIO y un valor 5.

La **ROM (Read Only Memory:** memoria de sólo lectura) es una memoria permanente —no puede modificarse, y al desconectar el ordenador no se borra. La ROM se usa para almacenar los programas especiales que dicen al ordenador cómo comportarse. Estos pueden incluir el **Sistema Operativo** que capacitará al CPU para dirigir la pantalla, el teclado y la memoria a la vez. Los sistemas operativos varían de una máquina a otra, aunque hay pocos estándares como el CP/M, MS-DOS y el Unix. La ventaja es que los programas de aplicación escritos en un ordenador usando un determinado sistema operativo, a priori pueden ser compatibles con cualquier otro ordenador que utilice el mismo sistema, sin tener en cuenta su diseño individual.

Hay otras clases de memorias, tales como **PROM, EPROM, EEPROM,** e incluso **EAROM,** cada una con sus propias características de realización, pero son usadas generalmente sólo en aparatos especiales o para propósitos de desarrollo anteriores a la producción a gran escala.

Todas las formas de memoria son medidas con las mismas unidades. Un **Bit** —abreviatura de **«Binary Digit»** (Dígito Binario)— es la unidad básica de la memoria de un ordeandor, y sólo puede ser usado para almacenar un «0» o un «1» —el equivalente a ser desconectado y conectado. Estos bits se combinan en unidades mayores para almacenar información útil. Ocho **bits** para un **byte** y muchos de los CPUs procesan un **byte** de una vez. Un **byte** es generalmente el equivalente a un dígito sencillo o un carácter, pero puedes representar algo más abstracto.

Algunos de los más nuevos y avanzados CPUs, como el 68000 y el Z8000, procesa 16 **bits** de una vez, un considerable aumento de velocidad. ¡Un proceso corriente puede ser llevado a cabo en aproximadamente la millonésima parte de un segundo!

1024 bytes hacen exactamente un **kilobyte** o **KByte** —por lo que un ordenador de 64 K tiene de hecho 64 * 1024 ó **65536 bytes** de memoria. La razón de este número tan raro es que puede ser convenientemente expresado en **«binario»,** que es el sistema de

Aunque para el principiante pueda ser desconcertante, la jerga es una parte necesaria del ordenador: tanto para evitar ambigüedades como para ahorrar el espacio, y esta sencilla guía para las expresiones más comúnmente usadas, ¡puede convertirle de novicio en experto!



numeración usado por los ordenadores.

El CPU es capaz únicamente de operaciones muy sencillas —como sumar dos dígitos— pero puede hacerlas a gran velocidad. Para que se ejecute alguna función real, el CPU necesita de una larga lista de instrucciones —un programa— escritas de una forma que pueda entender; esto es llamado **código máquina.**

Desgracidamente, es bastante difícil su manejo para los humanos, y consiste de filas de dígitos; el resultado es una programación muy lenta y que no admite errores. Por tanto, se han desarrollado para ordeandores varios lenguajes de alto nivel, que son bastante parecidos al inglés, y consecuentemente más fáciles de programar. **FORTRAN** es el lenguaje adecuado para los cálculos de ingeniería, mientras que la moyoría de los programas de gestión están escritos en **COBOL.** Con mucho, el lenguaje más común y también más fácil de aprender es el **BASIC.** El más estándar es el llamado Microsoft **BASIC** y se encuentra en el **PET, APPLE** y muchos otros.

PASCAL está alcanzando una gran popularidad. Es más duro de aprender pero mejor para escribir programas largos e inducir a una buena disciplina de programación a los nuevos programadores.

Los programas escritos en un lenguaje de alto nivel han de ser transformados en **Códigos Máquina** antes de que puedan ser programados por el **CPU.** Hay dos tipos de programas especiales que llevan a cabo esta traducción, el primero es el **«compilador»** que traduce todo el programa al código máquina antes de ejecutarlo, lo almacena en otro espacio de la memoria y después lo ejecuta.

El segundo tipo de **Basic** es el **«interpretado»** que actúa almacenado el programa en **Basic** y cada vez que lo ejecutamos es traducido a código máquina línea por línea.

Generalmente, un programa «interpretado» tardará unas 10 veces más que uno compilado, pero usará menos memoria porque sólo se almacena una versión.

Los «intérpretes» son más útiles si está desarrollando un programa debido a que al descubrir un error —un **BUG** en la jerga de los ordenadores— no se necesita recompilar todo el programa. La diferencia exacta entre «Interpretar» y «Compilar» requiere una mayor explicación.

Si un ordenador está diseñado para manejar un número determinado de lenguajes de programación, los diferentes «intérpretes» y «compiladores» estarán almacenados en la **RAM.** Las máquinas que sólo programen **BASIC,** sin embargo, tienen generalmente un «intérprete» en la **ROM.**

A esta clase de **software,** permanentemente almacenada en el **Hardware,** se la denomina a veces **¡FIRMWARE!** ¡Y ahora de vuelta al **Hardware!**

Los **circuitos integrados** (IC) se mantienen en una «placa de circuito impreso», que es una tarjeta rígida con muchas y finas pistas de cobre que interconectan a los componentes. Realmente algunos ordenadores baratos sólo poseen una placa de circuito impreso. De los extremos de la placa de circuito impreso sobresalen los **INTERFACES** que permiten al ordeandor intercambiar información con otros aparatos electrónicos. Este intercambio es conocido por **INPUT/OUTPUT** o I/0 (entrada/salida). Un **interface** está compuesto de varias líneas o cables —llamados bus— para manejar los datos (data) y de diferentes señales de control necesarias para sincronizar el tiempo entre varias operaciones. Las secuencias de los acontecimientos previos relacionadas con el intercambio de un conjunto de datos son conocidas por el nombre de protocolo —una expresión que proviene sin duda de la danza ritual de la diplomacia. Esta broma terminológica continúa en una etapa siguiente con una particular secuencia de salidas en un **interface** y se la conoce por **«handshake»** (apretón de manos», esto significa que la transacción está completa.

Hay varios **interfaces** comunes, así como un buen número de ellos que son propios y oscuros. A menos que use un medio especial de conversión, sólo puede interconectar aparatos que tengan un mismo **interface.**

En **Centronic, IEEE** 488, el **RS232** y el **S-100** son algunos de los estándares más ampliamente aceptados. Los dos primeros son **«buses»** paralelos, lo que significa que **8-bits** son transmitidos simultáneamente a lo largo de ocho líneas.

Los periféricos más comunes se nombran a continuación:

Las **Unidades de Cassetes,** que guardan los programas y datos de la memoria del ordenador en cassette, de modo que se puedan volver a cargar de nuevo cuando se necesiten.

El **«Disk Drive»** hace lo mismo utilizando una pieza circular de material magnético que gira dentro de una funda de cartón o plástico que se llama **Floppy Diskette.** Estos tienen gran capacidad de almacenamiento y permiten un acceso más rápido.

Se utiliza una **Impresora** para producir copias de la información en papel, y generalmente ofrecerá o una gran velocidad o una impresión de alta capacidad.

Por último, un **Modem** es una pequeña unidad que convierte los datos electrónicos en sonidos audibles y a la inversa, de forma que se pueda enviar la información a través de una línea telefónica ordinaria.

LOS PREMI

CADENAS HIFI-TS-55

El día 15 de enero de 1986, en presencia del notario de Madrid, don Angel Pérez Fernández, se efectuó el sorteo de los sistemas musicales Amstrad TS-55, entre los suscriptores.



GANADORES DE LOS SISTEMAS M

- Doña Carmen Pérez Vázquez
  Montevideo, local 12
  21005 HUELVA

- Don Francisco Javier Delgado Varela
  Villasandino, 9, 3.°, 2.°
  28011 MADRID

- Don José Miguel López López
  Avda. Carmen Miranda, 8, 2.° Izq.
  PRAVIA (Asturias)

- Doña Magdalena Olivera
  Rafael Calvo, 7, 4.° D
  28010 MADRID

- Don Ricardo Bajo Gámez
  Pedro Muguruza, 8
  28036 MADRID

- Doña Mercedes Ibáñez de Gracia
  Madrid, 67, 2.° Izq.
  GETAFE (Madrid)

OS LLEGAN

IMPRESORA



Don Juan Carlos Mira Navarro de LINEA 21, S. A., de Alicante, realiza la entrega del primer premio (una impresora DMP-1) del sorteo AMSTRADIEZ del mes de octubre a don Vicente Sanchis Verdu.

S MUSICALES AMSTRAD TS-55

- Don José Luis Ordóñez Ausín
  San Agustín, 17
  09001 BURGOS
- Don Angel Martínez Núñez
  Aprestadora, 18, 4.°-4.°
  HOSPITALET (Barcelona)
- Don Francisco Javier López Herrero
  Calderón de la Barca, número 5, 6.°B
  ALCALA DE HENARES (Madrid)
- Don Angel Vázquez González
  Apartado 46
  Grao de Gandía
  GANDIA (Valencia)

005883269

CLASE 8ª

ACTA DE SORTEO

RSX

CARACTERES EN TECNICOLOR

Si no, tendría que construir varios caracteres que, superpuestos, formaran las diferentes partes del caracter deseado e imprimirlos, con plumas diferentes, en modo transparente, uno encima de otro. Por ejemplo, si quiere un hombre con cuerpo azul, pelo rojo y cara blanca con un ojo, necesitaría un carácter para el cuerpo, otro para el pelo, y otro para la cara con un agujero para mostrar el color del fondo a través del ojo; ¡y todo esto trabajando en modo 1! Si fuera en modo O podría necesitar imprimir hasta 17 caracteres diferentes, lo cual resulta terriblemente lento.

Así pues, para aquellos que quieran un método fácil y rápido de imprimir caracteres multicolor (que, efectivamente, son sprites muy pequeños) en la pantalla sin recurrir a «montones» de código máquina, hemos diseñado dos sencillos RSXs. IMPUT para imprimir cada carácter, o una cadena de caracteres multicolor en la última posición de impresión de texto, y IGPUT para imprimir tanto un carácter como una cadena de ellos en la posición del cursor de gráficos.

Para utilizar cualquiera de los dos, introduzca simplemente el comando seguido de una serie de números, separados por comas. Por ejemplo:

```
10 LOCATE 10, 10: IMPUT,0,1,2,1
20 MOVE 100,100: IGPUT,4
30 MOVE 200,40:IGPUT,5,6
40 LOCATE 20,3: IMPUT,7
```

Los números escritos tras los comandos no se refieren al juego de caracteres normal, sino a otro que será diseñado separadamente y sobre el cual hablaremos más adelante.

Como puede ver, ambas rutinas son sencillas de usar; sin embargo, tienen algunas limitaciones.

El comando IMPUT no sitúa el cursor de texto detrás del último carácter que imprime, así que si desea mezclar caracteres multicolor y normales debe utilizar un comando LOCATE detrás de cada IMPUT. El comando IGPUT sitúa el caracter en la posición de byte más cercana en la pantalla, lo cual no es siempre la posición de pixel. Esto significa que puede moverse arriba y abajo de la pantalla con bastante facilidad pero sólo se moverá de 8 en 8 bits de pantalla a la derecha o izquierda. Esto limita el movimiento horizontal a medio carácter en modo 1 (cuatro pixels) y un cuarto de carácter en modo 0 (dos pixels). En modo 0 esto no constituye una limitación excesiva pero si piensa que el movimiento será muy desigual en modo 1, pruebe entonces a diseñar dos o más caracteres con la forma que quiere ligeramente modificada en cada uno, de modo que, imprimiéndolos uno detrás de otro consiga un movimiento suave. Esta es la técnica utilizada en el programa de demostración.

PAGUE EL PRECIO

Hay que pagar un precio a cambio de toda esta facilidad de uso: se necesita más memoria para cada carácter a imprimir. En lugar de los 8 bytes habituales, un carácter multicolor en modo 1 necesita 16 bytes, y uno en modo 0, 32 bytes. Pero esto no es tan malo si considera que habría tenido que utilizar hasta tres caracteres en modo 1 o 15 en modo 0 para conseguir el mismo resultado, y además mucho más lento. Un carácter en modo 2 sigue utilizando sólo 8 byes y la rutina funciona de hecho con el modo 2, aunque hay un pequeño problema para comprobarlo, ya que no puede tener más que una tinta, y debería utilizar en su lugar el método normal del BASIC.

UTILIZANDO EL PROGRAMA

Primero, teclee el programa 1, sálvelo y ejecútelo. Si ha cometido algún error en los DATAS, el programa le avisará en qué línea se encuentra. Cuando el programa funciona sin errores, sálvelo y reinicialice la máquina apagándola y encendiéndola de nuevo. Ahora teclee el programa 2 y sálvelo. Este le permitirá diseñar sus caracteres en cualquier modo que desee y salvarlos en cinta (o disco).

El programa diseñador es muy corto y simple. Después de decirle el modo que deseamos, entra en él. Para desplazar el cursor por el carácter, utilice las teclas de cursor, y use la tecla copy para poner y quitar la pluma. Para cambiar la pluma pulse la tecla [P] e introduzca el nuevo número de pluma. Para almacenar el carácter en memoria pulse la tecla [S] y déle un número. Puede tener todos los que quiera mientras recuerde cuánta memoria gasta cada uno. Para reeditar un carácter introducido antes en memoria, pulse [R] e introduzca su número. Cuando haya terminado de diseñar caracteres pulse [Q] y, después de preguntarle si está seguro, le preguntará cuántos salvar. Recuerde, el número de Sprites comienza en 0, de modo que de 0 a 7 equivale a 8 Sprites.

Ahora, para ver el fruto de su trabajo, debe reiniciar la máquina y ejecutar este programa.

```
10 IMPUT «MODO»;MM:MD = 1:IF
MM <> 1 THEN MD = 2
```

¿Cómo conseguir caracteres multicolor en la pantalla, tanto en coordenadas LOCATE como en coordenadas PIXEL? Hasta ahora tenía que hacerlo en código máquina, lo cual resulta un tanto excesivo si el 99 por 100 del programa funciona suficientemente bien y con la rapidez necesaria en BASIC.

```
10 'Programa 1
20 'RSX para impresion multicolor
40 '
50 h=HIMEM-1:IF PEEK(0)=0 THEN 70
60 POKE 0,0:h=h-192:MEMORY(h-1)
70 lin=1000:direc=h:FOR t=1 TO 192 STEP 8
80 suma=0:FOR b=0 TO 7
90 READ a$:a$=UPPER$(a$):n=1
100 IF LEN(a$)<>2 THEN 370
110 b$=MID$(a$,n,1)
120 GOSUB 350:IF a=0 THEN 370
130 n=n+1:IF n=2 THEN 120
140 p=VAL("&"+a$):POKE direc,p
150 direc=direc+1:suma=suma+p
160 NEXT b:READ a
170 IF a<>suma THEN 370
180 PRINT"LINEA";lin;"OK":lin=lin+10
190 NEXT t
200 r=h+9:a=INT(r/256):b=r-(a*256)
210 POKE h+1,b:POKE h+2,a
220 r=h+17:a=INT(r/256):b=r-(a*256)
230 POKE h+4,b:POKE h+5,a
240 r=h+21:a=INT(r/256):b=r-(a*256)
250 POKE h+9,b:POKE h+10,a
260 r=h+30:a=INT(r/256):b=r-(a*256)
270 POKE h+41,b:POKE h+42,a
280 POKE h+63,b:POKE h+64,a
290 r=h+31:a=INT(r/256):b=r-(a*256)
300 POKE h+122,b:POKE h+123,a
310 r=h+194:a=INT(r/256):b=r-(a*256)
320 POKE h+31,b:POKE h+32,a
330 CALL h
340 END
350 a=INSTR("0123456789ABCDEF",b$)
360 RETURN
370 PRINT "ERROR EN LA LINEA";lin;"!":END
1000 DATA 01,29,4E,21,31,4E,C3,D1, 684
1010 DATA BC,35,4E,18,18,00,18,11, 408
1020 DATA 00,00,00,00,00,4D,50,55, 242
1030 DATA D4,47,50,55,D4,00,00,30, 708
1040 DATA 75,F5,AF,18,03,F5,3E,01, 872
1050 DATA 32,3E,4E,F1,A7,C8,47,F5,1114
1060 DATA DD,23,DD,23,10,FA,CD,11,1000
1070 DATA BC,47,3E,02,90,F5,3A,3E, 832
1080 DATA 4E,A7,28,0A,CD,78,BB,2D, 852
```

(Continúa)

```
20 INPUT «CUANTOS SPRITES»;S:MODE MM
30 S = S*16*MD
40 H = HIMEM - 1
50 MEMORY (H-S)
60 H = HIMEM + 1
70 LOAD «SPRITES»,H
```

que cargará sus Sprites.

Ahora puede cargar y ejecutar el programa 1 de nuevo. Este moverá HIMEM y se pondrá a sí mismo por encima y se activará. Ya puede utilizar los RSXs y ver sus Sprites aparecer en la pantalla. No olvide que el comando IMPUT no moverá la posición del cursor, así que si introduce:

IMPUT,0,0,0

el mensaje «Ready» sobreescribirá sus Sprites, así que añada PRINT al final del comando. Y recuerde situar el cursor de gráficos en la pantalla con el comando MOVE para poder ver los caracteres, como:

MOVE 100,100:IMPUT,0

Bien, ahora debería tener las rutinas funcionando. Hay tres métodos para imprimir caracteres. Puede escribir los caracteres y borrarlos con IMPUT o

```
1090 DATA 25,CD,1A,BC,18,1B,CD,C6, 910
1100 DATA BB,B7,CB,1C,CB,1D,CD,11,1055
1110 DATA BC,F1,F5,A7,28,08,47,B7,1143
1120 DATA CB,1A,CB,1B,10,F9,CD,1D, 958
1130 DATA BC,F1,C1,C5,F5,DD,2B,DD,1549
1140 DATA 2B,4F,06,03,80,47,79,E5, 680
1150 DATA ED,5B,3F,4E,DD,6E,00,26, 838
1160 DATA 00,29,10,FD,19,EB,E1,87, 930
1170 DATA 20,01,3C,47,C5,E5,06,08, 604
1180 DATA 3A,05,00,A7,28,09,FE,01, 534
1190 DATA 28,0B,FE,02,1A,18,08,1A, 391
1200 DATA A7,28,05,18,02,1A,AE,77, 557
1210 DATA CD,26,BC,13,10,E2,E1,CD,1122
1220 DATA 20,BC,C1,10,D7,F1,C1,10,1094
1230 DATA B2,C9,00,00,00,00,00,00, 379
```

IGPUT con un carácter blanco sobre ellos. Debe reservar uno de sus caracteres para hacer esto. También puede utilizar el modo XOR para escribir los caracteres en la pantalla, con lo cual se combinarán con lo que tenga en la pantalla, en la misma forma que en juegos como «Manic Miner» y «Sorcery». Puede borrarlo reimprimiendo el mismo carácter. El tercer método sólo imprimirá las partes del carácter que tengan alguna tinta. No imprime bytes cero. Supongamos, por ejemplo, que sólo desea un carácter tamaño «modo 1» pero en modo 0. Si sólo diseña medio carácter, digamos la mitad izquierda, y deja el resto en blanco, cuando lo imprima junto a otro obtendrá normalmente el extraño efecto de borrar lo que había en la pantalla sin que aparezca nada en su lugar. Este método de imprimir suprime este efecto y permite imprimir caracteres pequeños sin un «halo» a su alrededor. Esto es casi igual al modo transparente normal. Esto significa que debe usarse el primer método de impresión si quiere borrar algo con un carácter en blanco; de otro modo no ocurrirá nada.

Para seleccionar los métodos, simplemente ejecute POKE 5,n donde n es:

2 a 255 para impresión normal.
1 para impresión XOR.
0 para impresión transparente.

Para ver a la rutina en acción teclee la primera parte del pequeño programa de demostración que prepara los caracteres. A continuación cargue y ejecute el programa 1 y después cargue y ejecute la segunda parte del programa de demostración para ver un balón moviéndose suavemente a través de la pantalla.

Como punto final, el programa funcionará en cualquier lugar de la memoria, así que si ha diseñado un juego de caracteres distinto con el comando SYMBOL AFTER hágalo antes de cargar y ejecutar los caracteres multicolores y el programa 1 y todo irá bien.

```
10 'Programa 2 -------- DISENADOR DE CARACTERES
20 INPUT "Modo";mm:md=1
30 IF mm<>1 THEN md=2
40 SPEED KEY 10,1:MODE mm
50 WINDOW#1,1,40/md,24,25
60 MEMORY 19999
70 q=1:BORDER 0
80 PEN 1:tt=TEST(x,398-y):LOCATE INT(x/(2*md))+10,INT(y/2)+1:PRINT CHR$(203)
90 k$=INKEY$:IF k$="" THEN 80
100 kk=ASC(k$)
110 LOCATE INT(x/(2*md))+10,INT(y/2)+1:PEN tt:PRINT CHR$(143):PEN q
120 x=x-(2*md)*((kk=&F3 AND x<(14*md))-(kk=&F2 AND x<>0))
130 y=y-2*((kk=&F1 AND y<14)-(kk=&F0 AND y<>0))
140 IF k$="p" OR k$="P" THEN PEN 1:INPUT#1,"Pluma";q:PEN q:CLS#1
150 IF kk=224 THEN te=1-(te=1):PEN 1:LOCATE 1,16:PRINT"OFF":IF te=1 THEN LOCATE
1,16:PRINT"ON "
160 LOCATE 1,15:PEN 1:PRINT"pluma ";q;"  ":PEN q
170 IF k$="r" OR k$="R" THEN 240
180 IF k$="s" OR k$="S" THEN 220
190 IF k$="q" OR k$="Q" THEN 280
200 IF te=1 THEN LOCATE INT(x/(2*md))+10,INT(y/2)+1:PRINT CHR$(143):PLOT x,398-y
,q
210 GOTO 80
220 PEN 1:INPUT#1,"Sprite";a:a=a*16*md+20000:PEN q:CLS#1
230 c=49152:FOR b=0 TO (2*md)-1:FOR n=0 TO 7:POKE a+8*b+n,PEEK(c+(n*2048)+b):NEX
T n:NEXT b:GOTO 80
240 PEN 1:INPUT#1,"Sprite";a:a=a*16*md+20000:PEN q:CLS#1
250 c=49152:FOR b=0 TO (2*md)-1:FOR n=0 TO 7:POKE c+(n*2048)+b,PEEK(a+8*b+n):NEX
T n:NEXT b:GOSUB 260:GOTO 80
260 FOR b=0 TO 14 STEP 2:FOR n=0 TO 14*md STEP(md*2):tt=TEST (n,398-b):LOCATE IN
T(n/(2*md))+10,INT(b/2)+1:PEN tt:PRINT CHR$(143):NEX
T n:NEXT b:PEN 1
270 RETURN
280 PEN 1:INPUT#1,"Salvar caracteres";r$:CLS#1:r$=UPPER$(r$):IF LEFT$(r$,1)<>"S"
 THEN 200
290 INPUT#1,"Cuantos";s:CLS#1
300 INPUT#1,"Con que nombre";nombre$:CLS#1:s=s*16*md:SAVE nombre$,B,20000,s:GOTO
 200
```

```
10 ;          M U L T I C O L O R
20 ;
30 ; Introduce los comandos residentes !MPUT y !GPUT
40 ;
50          LD   BC,COMAND            ;direccion de la tabla de comandos
60          LD   HL,BYTES             ;direccion de 4 bytes para el S.O.
70          JP   #BCD1
80 COMAND:  DEFW #A5CF                ;direccion de la tabla de nombres
90          JR   ENTRY2               ;salto a la rutina "MPUT"
100         NOP
110         JR   ENTRY1               ;salto a la rutina "GPUT"
120         DEFB #00
130 BYTES:  DEFB #00,#00,#00,#00      ;area para el uso del S.O.
140         DEFB "M","P","U",#D4      ;'MPU','T+#80'
150         DEFB "G","P","U",#D4      ;'GPU','T+#80'
160         DEFB #00                  ;marcador FIN DE TABLA
170 FLAG:   DEFB #00                  ;indica si se esta procesando MPUT o
180 ;                                 ;GPUT
190 DIRDAT: DEFW #A67C                ;direccion datos caracteres
200 ;
210 ;aqui comienza la rutina GPUT
220 ;
230 ENTRY1: PUSH AF                   ;preserva no. de caracteres
240         XOR  A                    ;borra acumulador
250         JR   COMUN
260 ;
270 ;aqui comienza la rutina MPUT
280 ;
290 ENTRY2: PUSH AF                   ;preserva no. de caracteres
300         LD   A,#01                ;si es MPUT, flag=1 - si es GPUT,
310 COMUN:  LD   (FLAG),A             ;flag=0
320         POP  AF                   ;recupera no. de caracteres
330         AND  A                    ;si no hay caracteres, retorna al BASIC
340         RET  Z
350         LD   B,A                  ;si si los hay, inicializa contador
360         PUSH AF                   ;salva no. de caracteres
370 BUCLE1: INC  IX                   ;y actualiza IX para que apunte
380         INC  IX                   ;detras del primer caracter
390         DJNZ BUCLE1
400         CALL #BC11                ;averigua el modo de pantalla
410         LD   B,A                  ;resta 2-modo
420         LD   A,#02
430         SUB  B
440         PUSH AF                   ;preserva el resultado
450         LD   A,(FLAG)             ;carga el flag (1: MPUT - 0: GPUT)
460         AND  A
470         JR   Z,GPUT2              ;si estamos en GPUT, salta
480         CALL #BB78                ;averigua posicion del cursor de texto
490         DEC  L                    ;convierte las coordnadas logicas
500         DEC  H                    ;a fisicas y averigua la direccion
510         CALL #BC1A                ;correspondiente de la memoria
520         JR   MPUT2                ;de pantalla
530 GPUT2:  CALL #BBC6                ;averigua coordenadas del cursor grafico
540         OR   A                    ;borra el acarreo
550         RR   H                    ;convierte coord. Y de "usuario"
560         RR   L                    ;a coord. Y de "base"
570         CALL #BC11                ;averigua el modo de pantalla
580         POP  AF                   ;recupera 2-modo
590         PUSH AF                   ;y lo vuelve a preservar
600         AND  A                    ;prueba si es 0 (modo 2)
610         JR   Z,PIXEL              ;si es modo 2, salta
620         LD   B,A                  ;si no, pasa 2-modo a B
630 BUCLE2: OR   A                    ;borra el acarreo
640         RR   D                    ;convierte coord. X de "usuario" a
650         RR   E                    ;coord. X de "base"
660         DJNZ BUCLE2
670 PIXEL:  CALL #BC1D                ;calcula direcc. pixel en pantalla
680 MPUT2:  POP  AF                   ;recupera 2-modo
690         POP  BC                   ;recupera no. de caracteres
700 OTRO:   PUSH BC                   ;salva no. de caracteres
710         PUSH AF                   ;salva 2-modo
720         DEC  IX                   ;apunta al caracter
730         DEC  IX
740         LD   C,A                  ;carga en C 2-modo
750         LD   B,#03
760         ADD  A,B                  ;le suma 3
```

(Continúa)

```
 770          LD   B,A           ;pasa 2-modo+3 a B
 780          LD   A,C           ;recupera 2-modo
 790          PUSH HL            ;preserva direccion del pixel
 800          LD   DE,(DIRDAT)   ;DE=#A67C principio datos
 810          LD   L,(IX+0)      ;de caracter
 820          LD   H,#00
 830 BUCLE3:  ADD  HL,HL         ;calcula direccion en que se halla
 840          DJNZ BUCLE3
 850          ADD  HL,DE
 860          EX   DE,HL         ;pasa direccion a DE
 870          POP  HL            ;recupera direccion de pixel
 880          ADD  A,A           ;duplica 2-modo para calcular el ancho
 890          JR   NZ,ANCHO      ;del caracter.Si es modo 2, ancho=1
 900          INC  A
 910 ANCHO:   LD   B,A           ;ancho de caracter a B
 920 COLUMN:  PUSH BC            ;preserva ancho y 2-modo
 930          PUSH HL            ;preserva direccion pixel
 940          LD   B,#08         ;inicializa contador
 950 BUCLE4:  LD   A,(#0005)     ;carga el modo de impresion
 960          AND  A             ;prueba si es cero (transparente)
 970          JR   Z,TRANSP
 980          CP   #01           ;prueba si es uno (XOR)
 990          JR   Z,MODXOR      ;si no, es modo normal
1000          CP   #02
1010          LD   A,(DE)        ;coge el byte del caracter
1020          JR   NORMAL
1030 TRANSP:  LD   A,(DE)        ;carga primer byte del caracter
1040          AND  A             ;si es cero, pasa a NORM2
1050          JR   Z,NORM2
1060          JR   NORMAL        ;y si no, a NORMAL
1070 MODXOR:  LD   A,(DE)        ;carga byte del caracter
1080          XOR  (HL)          ;XOR con el de la pantalla
1090 NORMAL:  LD   (HL),A        ;envia byte a la pantalla
1100 NORM2:   CALL #BC26         ;calcula direccion byte inferior de
1110 ;                           ;la pantalla
1120          INC  DE            ;apunta al siguiente byte del caracter
1130          DJNZ BUCLE4        ;repite para los 8 bytes
1140          POP  HL            ;recupera direccion esquina superior
1150 ;                           ;del caracter
1160          CALL #BC20         ;calcula direcc. siguiente columna
1170          POP  BC            ;recupera ancho del caracter
1180          DJNZ COLUMN        ;repite hasta completar el caracter
1190          POP  AF            ;recupera 2-modo
1200          POP  BC            ;recupera no. de caracteres
1210          DJNZ OTRO          ;repite para todos los caracteres
1220          RET                ;vuelve al BASIC
```

```
10 'demo ......... parte 1
20 h=HIMEM
30 h=h-64
40 MEMORY (h-1)
50 FOR n=0 TO 15:READ a:POKE h+n,a:NEXT
60 FOR n=16 TO 31:POKE h+n,0:NEXT
70 FOR n=32 TO 55:READ a:POKE h+n,a:NEXT
80 FOR n=56 TO 63:POKE h+n,0:NEXT
90 DATA 48,115,247,247,243,240,112,48
100 DATA 192,224,248,248,240,240,224,192
110 DATA 0,16,48,48,48,48,16,0
120 DATA 240,240,240,240,243,247,247,243
130 DATA 0,128,192,192,200,200,128,0
```

```
10 'demo ........ parte 2
20 POKE 5,1:REM modo XOR
30 FOR n=0 TO 640 STEP 8
40 MOVE n,200:|GPUT,0,1
50 GOSUB 140
60 |GPUT,0,1:|GPUT,2,3
70 GOSUB 140
80 |GPUT,2,3
90 NEXT
100 FOR n=0 TO 400 STEP 2
110 MOVE 300,n:|GPUT,0
120 GOSUB 140:|GPUT,0
130 NEXT:GOTO 20
140 FOR z=0 TO 30:NEXT
150 RETURN
```

COMPRESOR D

CONSIGA MAS DE SU AMSTRAD COMPRIMIENDO LOS DATOS



l problema de salvar un volcado de pantalla en el CPC-464 estriba en que requiere 16 K de memoria para salvar cualquier tipo de imagen, tanto texto como texto y gráficos mezclados. Con un cassette el problema es la cantidad de tiempo que lleva cargar de nuevo la pantalla en memoria, aun cuando la velocidad de cassette es muy razonable si la comparamos con la de otros micros. La velocidad no es problema si posee un sistema de disco, pero con 16 K cada vez, su disco no tardará en llenarse.

COMPRESOR es un programa en código máquina diseñado para reducir la cantidad de memoria necesaria para albergar toda la información requerida para reproducir la imagen original. Esto se realiza en tres pasos:

1. Los valores actuales del MODO, OFFSET DE PANTALLA, COLOR DE BORDE y COLORES DE TINTAS se copian en cinta. A ser posible es preferible asegurarse de que el OFFSET DE PANTALLA es cero (ejecutando un comando MODE), o puede encontrarse una pequeña parte del display (una línea) diferente de cuando la salvó.

2. Las 200 líneas horizontales se comparan para ver si algunas coinciden.

3. Las 80 posiciones de memoria para cada una de las 200 líneas se inspeccionan para ver si existe alguna secuencia de números repetida.

Los tres programas que se muestran en los listados 1, 2 y 3 funcionan de la siguiente manera:

Listado 1: Programa Compresor

Este es el programa que realiza todo el trabajo duro de verificar y comprimir. La línea 10 fija la variable del sistema MEMORY a 5000 hexadecimal, para habilitar suficiente espacio para los datos comprimidos que se almacenan desde 9000 (Hex.) hacia abajo. Es posible que los datos no alcancen las posiciones cercanas a 5000 H pero es necesaria esta previsión por si lo hicieran. El programa en código máquina se encuentra a partir de 9030 H. Las posiciones de memoria entre 9000 H y 9030 H se utilizan para almacenar la información sobre el Modo y los colores de Borde y de las Tintas. El programa en código máquina está dividido en nueve secciones, lo cual le facilitará el trabajo de teclearlo. Cada sección tiene también un número de verificación para asegurarse de que los DATAS son correctos. Si se introduce algún DATA equivocado el programa BASIC le dirá en qué sección está el error.

La sección 1 limpia los diversos buffers que se van a utilizar inicializándolos con ceros. La sección 2 comprueba las 16000 posiciones de memoria para descubrir qué números se emplean en la pantalla. Esto se consigue inicializando a cero un bloque de 256 bytes de memoria y usando el número de la pantalla como el byte menos significativo de la dirección del bloque. Al final de la comprobación, todos los números del bloque que permanezcan a cero son por lo tanto los no utilizados en la pantalla.

La sección 3 inspecciona este bloque para contar cuántos números libres hay disponibles. El motivo de todo esto es que el programa COMPRESOR necesita dos números, que utiliza como «marcadores», para señalizar las líneas que son iguales y las secuencias de números que son iguales. Obviamente si todos los 256 números de 8 bits se utilizan en la pantalla no hay posibilidad de utilizar ninguno como marcador.

Casi la única posibilidad de que esto ocurra es si usted pokea deliberadamente los 256 números en la memoria de pantalla. Incluso con los dibujos más complejos, existen normalmente algunas docenas de números utilizables. Si no se puede conseguir ningún marcador, el programa retorna al BASIC, continuando si no con la sección 4.

La sección 4 produce en primer lugar un BEEP para informarle de que dispone de dos marcadores. Estos se encuentran en las posiciones de memoria 9001 H y 9002 H.

A continuación, la sección 5 recoge los valores actuales del OFFSET DE PANTALLA, MODO, COLOR DE BORDE y COLORES DE TINTAS, y los almacena entre 9004 H y 902 CH.

Las secciones 6 y 7 crean una pareja de «Bloques» que contienen los datos auxiliares para la inspección de las líneas de la pantalla.

La sección 8 compara las 200 líneas de pantalla para ver si algunas son iguales. Las líneas que sean iguales utilizarán sólo dos bytes: el número «marcador» seguido por el número de la línea similar.

DE PANTALLAS

La sección 9 comprobará a continuación cada línea individual de 80 bytes (a no ser que comience con el marcador "igual que") y buscará cualquier repetición de números. Si hay más de tres números consecutivos iguales se representan como:

Segundo "número marcador".

Cantidad de números iguales.

Valor del número que se repite.



LISTADO 1: PROGRAMA COMPRESOR

```
10 MEMORY &5000:posicion=&9030:E$="ERROR
 EN "
20 GOSUB  500:IF suma<>1528 THEN PRINT E
$;"1"
30 GOSUB  500:IF suma<>1320 THEN PRINT E
$;"2"
40 GOSUB  500:IF suma<>2488 THEN PRINT E
$;"3"
50 GOSUB  500:IF suma<>1831 THEN PRINT E
$;"4"
60 GOSUB  500:IF suma<>5360 THEN PRINT E
$;"5"
70 GOSUB  500:IF suma<>1611 THEN PRINT E
$;"6"
80 GOSUB  500:IF suma<>806 THEN PRINT E$
;"7"
100 GOSUB 500:IF suma<>10394 THEN PRINT
E$;"8"
120 GOSUB 500:IF suma<>13339 THEN PRINT
E$;"9"
199 END:'******************************
***********
500 suma=0:codigo$="":READ codigo$
510 WHILE codigo$<>"-1":numero=VAL("&"+c
odigo$)
520 POKE posicion,numero:posicion=posici
on+1
530 suma=suma+numero:READ codigo$:WEND:R
ETURN
999 END:'******************************
***********
1000 '1***********INICIALIZA todos los B
UFFERS a ZERO***********************
1010 DATA 21,0,90,3e,0,77,23,77,23,77,1,
0,4,21,0,93,36,0,23,b,78,b1,20,f8,-1
1020 '----------------------------------
-----------------------------------
2000 '2*************** BUSCA los NUMEROS
 LIBRES ***************************
2010 DATA 1,0,40,21,0,c0,16,93,7e,5f,12,
23,b,78,b1,20,f7,-1
2020 '----------------------------------
-----------------------------------
3000 '3**************** RETORNA si hay p
ocos PUNTEROS **********************
3010 DATA 16,0,21,ff,93,6,ff,7e,fe,0,20,
1,14,2b,10,f7,7a,fe,0,c8,fe,1,c8,-1
3020 '----------------------------------
-----------------------------------
4000 '4*************** PASA los MARCADOR
ES al BUFFER..9001/3 ***************
4010 DATA 3e,7,cd,5a,bb,21,1,93,6,2,11,1
,90,7e,fe,0,20,6,7d,12,13,5,28,3,23
4020 DATA 18,f2,-1
4030 '----------------------------------
-----------------------------------
```

Como se utilizan tres bytes, sólo merece la pena hacerlo si hay una repetición de más de tres números. Por ejemplo, una línea completa de 80 números iguales —digamos 99— se comprimiría como 12 80 99, suponiendo que 12 representa al segundo marcador.

Esta sección también almacena estos números y marcadores desde 9000 H hacia abajo. La dirección de la última posición de memoria utilizada se almacena en 9008/9 H para permitir el cálculo de la longitud del fichero comprimido. Como bien puede observar, el programa ha de realizar mucho trabajo, motivo por el cual se utiliza el código máquina.

Listado 2: Programa Descompresor

Comparado con el programa para comprimir los datos de la pantalla, el programa para descomprimir estos datos es relativamente simple. El MODO, OFFSET DE PANTALLA y COLORES DE TINTAS se fijan primero. Entonces se inspeccionan los números almacenados desde 9000 H hacia abajo. Si se encuentra un marcador se ejecuta la acción correspondiente con los números siguientes; en caso contrario el número es copiado en la memoria de la pantalla. El programa en código máquina del listado 2 se almacena a partir de 9200 H.

Listado 3

Este programa se ha diseñado para utilizarlo como una subrutina para calcular la longitud del fichero comprimido, y para salvarlo a cinta como explicaremos brevemente.

Utilización de los programas

Teclee el listado 1 y cuando el programa funcione sin errores en los datas sálvelo en cinta. Ejecute el programa BASIC que pokeará el código máquina en memoria. El programa BASIC puede ser borrado con el comando NEW, ya que no se le necesitará. Es interesante, sin embargo, conservar una copia de seguridad en cinta.

Teclee el listado 2, sálvelo, ejecútelo y bórrelo (NEW). Ahora ya está listo para ver si funciona el programa.

Ajuste el OFFSET a cero tecleando MODE 0, MODE 1 o MODE 2. A continuación escriba o dibuje algo en la pantalla, cuidando de no provocar un SCROLL de pantalla, ya que esto cambiaría el actual valor 0 del OFFSET. Para salvar el dibujo de la pantalla como fichero comprimido teclee CALL &9030: si todo va bien escuchará un BEEP y aparecerá el mensaje READY pasados de uno a diez segundos, dependiendo de la complejidad de la pantalla.

Para verificar el correcto funcionamiento del programa decodificador (listado 2) cambie primero el MODO y el BORDE y las TINTAS si lo desea, o simplemente borre la pantalla, y escriba CALL &9200.

Si todo va bien, su dibujo original aparecerá en un segundo. Si el programa no funciona correctamente verifique todos los datas en los listados para comprobar que los ha introducido correctamente. Las sumas de verificación no son infalibles; después de todo, pudo introducir dos valores intercambiados. La suma será la misma, pero los dos valores intercambiados en el programa en código máquina producirán el desastre. Cuando los dos programas funcionen, introduzca el listado 3 y salve el fichero comprimido a cinta (o disco). Si solamente está creando y salvando dibujos a cinta entonces no es necesario que el programa del listado 2 esté en memoria. Análogamente, si está cargando los ficheros en memoria para devolverlos a la pantalla, no es necesario que el programa del listado 1 esté en memoria. Ambos pueden funcionar perfectamente tanto juntos como separados, según sea necesario.

Supongamos que usted tiene un programa en BASIC para crear hermosos dibujos que desea salvar en forma comprimida. Esto es lo que neccesitará hacer:

En primer lugar, cargue el listado 1 y ejecútelo para cargar el programa en código máquina en memoria. Cargue su programa BASIC para crear los dibujos y a continuación mezcle (MERGE) la subrutina del listado 3 con su

```
5000 '5***************** COPIA MODO, OFFS
ET Y COLORES ************************
5010 DATA cd,b,bc,22,5,90,21,0,0,cd,5,bc
,cd,11,bc,32,4,90,21,a,90,e5,cd
5020 DATA 3b,bc,e1,70,23,71,23,3e,10,f5,
e5,cd,35,bc,e1,f1,70,23,71,23,3d
5030 DATA fe,0,20,f0,-1
5040 '-----------------------------------
------------------------------------
6000 '6***************** CREA BLOQUE 400
de MEMORIA DE PANTALLA **************
6010 DATA 6,c8,11,0,94,21,0,c0,7d,12,13,
7c,12,13,cd,26,bc,10,f5,-1
6020 '-----------------------------------
------------------------------------
7000 '7***************** CREA BLOQUE 200
************************************
7010 DATA 21,0,93,6,c8,75,23,10,fc,-1
7020 '-----------------------------------
------------------------------------
8000 '8***************** COMPARA LAS LINEA
S **********************************
8010 DATA 21,0,93,e5,21,1,93,dd,21,0,94,
fd,21,2,94,dd,46,1,dd,4e,0,fd,56,1
8020 DATA fd,5e,0,7d,be,20,11,e5,c5,e1,6
,50,1a,be,20,9,13,23,10,f8,e1,c1,71
8030 DATA c5,e5,e1,7d,fe,c7,20,16,e1,7d,
fe,c6,28,17,dd,23,dd,23,23,7d,be,20
8040 DATA f2,e5,dd,e5,fd,e1,18,c2,fd,23,
fd,23,23,18,bb,-1
8050 '-----------------------------------
------------------------------------
9000 '9***************** COMPRIME LAS LINE
AS *********************************
9010 DATA 11,0,90,3e,1,12,1b,1,0,c8,dd,2
1,0,94,fd,21,0,93,d5,dd,66,1,dd,6e,0
9020 DATA fd,7e,0,b9,20,62,c5,1,50,0,11,
0,96,ed,b0,3a,1,90,32,50,96,c1,d1,c5
9030 DATA 21,0,96,3e,0,32,7,90,6,1,7e,23
,4f,3a,1,90,be,28,32,79,be,20,3,4
9040 DATA 18,f1,4f,78,fe,3,28,15,fe,2,28
,11,fe,1,28,d,3a,2,90,12,1b,78,12,1b
9050 DATA 79,12,1b,18,5,79,12,1b,10,fc,3
a,7,90,fe,63,28,c,18,c5,18,9c,3e,63
9060 DATA 32,7,90,79,18,cd,c1,18,b,d1,3a
,1,90,12,1b,fd,7e,0,12,1b,dd,23,dd,23
9070 DATA fd,23,c,10,df,ed,53,8,90,c9,-1
```

LISTADO 2: PROGRAMA DESCOMPRESOR

```
10 MEMORY &5000:posicion=&9200
1000 suma=0:codigo$="":READ codigo$
1010 WHILE codigo$<>"-1":numero=VAL("&"+
codigo$)
1020 POKE posicion,numero:posicion=posic
ion+1
1030 suma=suma+numero:READ codigo$:WEND
```

LISTADO 2: PROGRAMA DESCOMPRESOR

```
1040 IF suma<>15781 THEN PRINT "Error en
 los DATAS !":END
13000 '****************** DESCOMPRIME LO
S DATAS (Call &9200) ******************
13010 DATA 3a,4,90,cd,e,bc,21,0,0,cd,5,b
c,21,a,90,46,23,4e,23,e5,cd,38,bc,e1
13020 DATA 3e,10,f5,46,23,4e,23,e5,cd,32
,bc,e1,f1,3d,fe,0,20,f0,21,ff,8f,dd,21
13030 DATA 0,90,11,0,c0,e,c8,d5,6,50,7e,
dd,be,1,28,34,dd,be,2,20,18,2b,c5,4e,2b
13040 DATA 7e,12,13,5,d,79,fe,0,20,f6,78
,c1,47,2b,fe,0,20,df,18,5,12,13,2b,10
13050 DATA d8,d1,d,79,fe,0,28,25,e5,62,6
b,cd,26,bc,54,5d,e1,18,c3,c5,2b,46,4,e5
13060 DATA 21,0,c0,cd,26,bc,10,fb,cd,29,
bc,1,50,0,ed,b0,e1,2b,c1,18,d4,2a,5,90
13070 DATA cd,5,bc,c9,-1
13080 '---------------------------------
---------------------------------------
```

—LISTADO 3—

```
10000 '*** CALCULA DIRECCION DE COMIENZO
 Y LONGITUD DE UN FICHERO COMPRIMIDO ***
10010 lsb=PEEK(&9008):msb=PEEK(&9009):re
sultado=msb*256+lsb
10020 total=36912-resultado:PRINT"Memori
a utilizada =";total
10030 PRINT"Pulse <ENTER> para SALVAR a
CINTA"
10040 PRINT"Cualquier otra tecla para FI
N"
10050 WHILE tecla$="":tecla$=INKEY$:WEND
10060 IF tecla$<>CHR$(13) THEN END
10070 LINE INPUT"Introduzca el nombre de
l fichero ";nombre$
10080 SAVE"!"+nombre$,B,resultado,total
10090 '---------------------------------
---------------------------------------
```

programa en BASIC. Inserte la siguiente línea en esta subrutina:

10005 CALL &9030

Cuando haya creado su dibujo entonces GOSUB 10000, introduzca el nombre que desee darle al dibujo y el fichero se salvará a cinta como fichero binario. Repita este proceso si necesita salvar más dibujos.

Para recuperar los dibujos cargue el programa del listado 2 y ejecútelo. Si ha salvado cinco dibujos en cinta entonces éstos pueden ser recuperados y mostrados uno tras otro utilizando un programa como este:

```
10 FOR PANTALLA=1 TO 5
20 LOAD "!"
30 CALL &9200
40 NEXT PANTALLA
```

Cada dibujo será cargado y mostrado en el monitor. Ahora bien, ¿cuánta memoria se ahorra?

Como media, el ahorro es de un 50 %, lo cual, en términos de ordenador es una economía considerable. Dibujos muy simples pueden utilizar sólo 1 K de memoria mientras que dibujos complejos pueden necesitar 12 K o más. Muchos de mis dibujos utilizan aproximadamente 8 K así que podrá albergar el doble de dibujos en su cinta o disco, y necesitará la mitad del tiempo habitual para cargarlos.

En algunos casos su programa para crear dibujos será mucho más pequeño que el fichero comprimido. Sin embargo, si la rutina de dibujo es del tipo complejo de tres dimensiones, utilizando muchas fórmulas matemáticas, el tiempo necesario para realizar el dibujo puede ser considerable (a veces horas) a pesar de que el CPC-464 no es perezoso. Este es el tipo de situaciones en que la técnica de compresión alcanza su máxima utilidad.

Angel ZARAZAGA

```
10 ;              PROGRAMA  DESCOMPRESOR
20 ;              --------  ------------
30
40          ORG  #9200
50 comien:  LD   A,(#9004)     ;coge MODO de pantalla
60          CALL #BC0E         ;fija MODO de pantalla
70          LD   HL,#0000      ;fija OFFSET de pantalla a cero
80          CALL #BC05
90          LD   HL,#900A      ;carga direccion de colores de BORDE
100         LD   B,(HL)        ;carga primer color de BORDE
110         INC  HL
120         LD   C,(HL)        ;carga segundo color de BORDE
130         INC  HL
140         PUSH HL            ;salva direccion de la tabla de tintas
150         CALL #BC38         ;fija colores de BORDE
160         POP  HL            ;recupera direccion
170         LD   A,#10         ;carga numero de plumas (16)
180 tintas: PUSH AF            ;salva el contador de plumas
190         LD   B,(HL)        ;carga colores de la pluma
200         INC  HL            ;correspondiente
210         LD   C,(HL)
220         INC  HL
230         PUSH HL            ;preserva puntero
240         CALL #BC32         ;fija colores
250         POP  HL            ;recupera puntero
260         POP  AF            ;recupera el contador de plumas
270         DEC  A             ;decrementa contador de plumas
```

```
280            CP    #00
290            JR    NZ,tintas          ;siguiente pluma hasta 16
300            LD    HL,#3FFF           ;principio del fichero comprimido
310            LD    IX,#9000           ;direccion de los marcadores
320            LD    DE,#C000           ;comienzo de la memoria de pantalla
330            LD    C,#C8              ;carga no. de lineas en la pantalla
340 contin:    PUSH  DE                 ;preserva direccion de pantalla
350            LD    B,#50              ;carga no. de bytes por linea
360 mark?:     LD    A,(HL)             ;coge dato del fichero
370            CP    (IX+1)             ;si es el primer marcador salta a la
380            JR    Z,mark1            ;rutina correspondiente
390            CP    (IX+2)             ;si no es el segundo marcador salta
400            JR    NZ,imp             ;a la rutina de impresion en pantalla
410
420 ;                  rutina  SEGUNDO MARCADOR
430
440 mark2:     DEC   HL
450            PUSH  BC                 ;salva contadores de bytes y lineas
460            LD    C,(HL)             ;carga numero de bytes repetidos
470            DEC   HL
480 repite:    LD    A,(HL)             ;carga valor del byte a repetir
490            LD    (DE),A             ;carga byte en pantalla
500            INC   DE                 ;siguiente posicion de pantalla
510            DEC   B                  ;decrem. contador no. de bytes/linea
520            DEC   C                  ;decrementa contador de bytes
530            LD    A,C                ;repite el numero de veces necesario
540            CP    #00
550            JR    NZ,repite
560            LD    A,B                ;final de linea?
570            POP   BC                 ;recupera contador de lineas (C)
580            LD    B,A
590            DEC   HL
600            CP    #00                ;si no, siguiente dato del fichero
610            JR    NZ,mark?           ;si es el final de linea, calcula
620            JR    otra               ;la direccion de la siguiente
630
640 ;                rutina de  IMPRESION DIRECTA
650
660 imp:       LD    (DE),A             ;imprime el byte en pantalla
670            INC   DE                 ;siguiente direccion de pantalla
680            DEC   HL                 ;siguiente dato del fichero
690            DJNZ  mark?              ;comprueba si es un marcador
700 otra:      POP   DE                 ;recupera direccion linea de pantalla
710            DEC   C                  ;actualiza contador lineas
720            LD    A,C                ;si no quedan lineas, fin del programa
730            CP    #00
740            JR    Z,fin
750            PUSH  HL                 ;preserva puntero datos del fichero
760            LD    H,D
770            LD    L,E
780            CALL  #BC26              ;calcula direccion siguiente linea
790            LD    D,H
800            LD    E,L
810            POP   HL                 ;recupera puntero datos fichero
820            JR    contin             ;continua
830
840 ;                  rutina  PRIMER MARCADOR
850
860 mark1:     PUSH  BC                 ;salva contadores de bytes y lineas
870            DEC   HL
880            LD    B,(HL)             ;carga no. de la linea igual
890            INC   B
900            PUSH  HL                 ;preserva puntero datos fichero
910            LD    HL,#C000           ;busca linea a copiar
920 bucle:     CALL  #BC26              ;SCR NEXT LINE
930            DJNZ  bucle
940            CALL  #BC29              ;SCR PREV LINE
950            LD    BC,#0050           ;copia la linea de (HL) a (DE)
960            LDIR
970            POP   HL                 ;recupera puntero fichero
980            DEC   HL                 ;siguiente dato del fichero
990            POP   BC                 ;recupera contadores bytes y lineas
1000           JR    otra
1010
1020 ;                   SALIDA DEL PROGRAMA
1030
1040 fin:      LD    HL,(#9005)         ;carga OFFSET de pantalla
1050           CALL  #BC05              ;fija OFFSET de pantalla
1060           RET
```



OFERTA ESPECIAL
2 PROGRAMAS DE OBSEQUIO

☐ **CONSIDEREME SUSCRIPTOR DE LA REVISTA AMSTRAD USER por un año (12 números)**

Recibiré en mi domicilio, como obsequio especial, **dos programas en cassette**

NOMBRE ____________ 1.º APELLIDO ____________ 2.º APELLIDO ____________

CALLE, AVDA., PLAZA ______________________

LOCALIDAD ____________ CODIGO POSTAL ______ PROVINCIA ____________

FORMA DE PAGO: ☐ CONTRA REEMBOLSO
☐ POR GIRO POSTAL
☐ POR TALON DE BANCO (1)
☐ CON TARJETA DE CREDITO

PRECIO SUSCRIPCION 3.100 PTAS.*

* Precio normal en quioscos: 3.600 ptas. anuales

Carguen 3.100 ptas. a mi tarjeta: AMERICAN EXPRESS ☐ VISA ☐
Núm. de mi tarjeta ______________________

Fecha de caducidad ______________________

Firma

(1) Dirigir a INDESCOMP, S. A.

A-5

AD-MF048 P.V.P. 19.500
impresora aleatoria. (Opcional-

AD-CG062 P.V.P. 19.500

AD-CG063 P.V.P. 24.500

AD-AF069 P.V.P. 19.500
os con totalización de compras, y albaranes. (Dos unidades de

AD-CP088 P.V.P. 15.500

6128" PM/Plus

ALMACEN-FACTURACION AD-AF025 P.V.P. 15.500

Control de entradas y salidas de inventarios con totalización de compras y ventas. Las facturas descuentan la mercancía del almacén automáticamente. Emite recibos, facturas y albaranes.
(Una unidad de disco).

RPA CLIENTES-PROVEEDORES-FACTURACION-MAILING AD-CP071 P.V.P. 15.500

Le permite tener a sus clientes y proveedores clasificados por sectores, sus totales facturados y su IVA. Totalización parcial y total. Permite hacer mailing y el control y seguimiento de la cartera de sus clientes.

PRA PROCETEXT AD-PT061 P.V.P. 9.500

Procesador de textos y mailing. Incluye márgenes, impresión, tabuladores, acentos, etc.

RPA AGENDA ROBOT AD-AR032 P.V.P. 12.500

Integrado por cuatro subprogramas: Agenda-Fichero de clientes, Dietario, Calculadora y Base de Datos.

CONTABILIDAD GENERAL II AD-CG027 P.V.P. 15.500

Contabilidad general española, según el plan contable. Incluye: Anotación de asientos, Modificación de apuntes, Asientos externos, Libro diario, Mayor, Balance, Fin de periodo, Fin de ejercicio. Con cálculos matemáticos realizados en ensamblador que garantizan alta precisión.

"AMSTRAD 6128" Serie Profesional 2000. CPM/Plus

RPA MULTI AGENDA ROBOT AD-MR072 P.V.P. 15.500

Realizada en Pascal.

RPA MULTIBASE 3 AD-MB046 P.V.P. 19.500

Permite hacer gráficos de curva, tarta e histogramas, lineal y en tres dimensiones.

RPA FACTURACION 6128 AD-FA047 P.V.P. 19.500

Incluye apuntes de contabilidad, formas de pago y diversas opciones de impresora: Factura, Albarán o Recibo. Con totalización parcial o general. Incluye IVA. (Opcionalmente dos unidades de disco).

RPA NOMINAS AD-NO045 P.V.P. 19.500

Comprende impresos TC1, TC2, apuntes de contabilidad e informes.

RPA GESTION DE EMPRESA AD-GE030 P.V.P. 65.000

Integración de los programas: Contabilidad General, Nóminas, Facturación, Clientes y Proveedores. (Dos unidades de disco).

RPA GEOGRAFIA DE ESPAÑA	AD-GS051	P.V.P. 9.500
RPA GEOGRAFIA UNIVERSAL	AD-GU073	P.V.P. 9.500
RPA CLIMATOLOGIA	AD-CL073	P.V.P. 9.500
RPA DEMOGRAFIA	AD-DE075	P.V.P. 9.500
RPA CUERPO HUMANO	AD-CH052	P.V.P. 9.500
RPA PLANETARIO	AD-PL076	P.V.P. 9.500
RPA TEOREMAS GEOMETRICOS	AD-TG077	P.V.P. 9.500

"AMSTRAD 6128" Sectores Profesionales. CPM/Plus

ARTICULO	REF.	P.V.P.
RPA MEDICO-CLINICO-FARMACOS	AD-CF028	15.500
RPA MEDICO-CLINICO-DENTISTA	AD-CE049	15.500
RPA CALCULO DE ESTRUCTURAS	AD-CE078	12.500
RPA VIDEO CLUB	AD-VC029	28.500

"AMSTRAD 8256" CPM/Plus

RPA ALMACEN-FACTURACION	AD-AF056	24.500
RPA CONTABILIDAD GENERAL 5	AD-GE055	24.500
RPA GESTION DE EMPRESA	AD-GE057	76.500
RPA MULTI AGENDA ROBOT	AD-MR085	19.500
RPA NOMINAS	AD-NO059	28.500
RPA MULTIBASE 3	AD-MB060	28.500
RPA FACTURACION 8256	AD-FA053	28.500
RPA MASTER FILE	AD-MF054	28.500
RPA VIDEO CLUB	AD-VC066	35.500
RPA GRAFICOS DE GESTION	AD-FG067	19.500
BASE DE DATOS	AD-ED080	24.500
"RPA ELEPHANT DATA BASE"		
RPA ADMINISTRACION DE FINCAS	AD-AD081	35.500
RPA PROJECT PLANNING	AD-PP082	19.500
RPA BRAINSTORM PLUS	AD-BS083	19.500
RPA CLIENTES-PROVEEDORES-FACT. MAILING	AD-CP084	24.500

Los programas para el Amstrad 8256 estarán disponibles en la última semana de enero. Sus características serán similares a los del Amstrd 6128, con mayor capacidad de datos y definición de pantalla. Todos ellos funcionarán con un sistema de protección (un Interfaz) denominado "Interproc".

Galileo, 25 - Entreplanta A - 28015-Madrid
Tels. 447 97 51 / 447 98 09

```
280
290
300
310
320
330
340 cont
350
360 mark
370
380
390
400
410
420 ;
430
440 mark
450
460
470
480 repi
490
500
510
520
530
540
550
560
570
580
590
600
610          JR   NZ,mark?              ;si es el final de linea, calcula
620          JR   otra                  ;la direccion de la siguiente
630
640 ;              rutina de  IMPRESION DIRECTA
650
660 imp:     LD   (DE),A                ;imprime el byte en pantalla
670          INC  DE                    ;siguiente direccion de pantalla
680          DEC  HL                    ;siguiente dato del fichero
690          DJNZ mark?                 ;comprueba si es un marcador
700 otra:    POP  DE                    ;recupera direccion linea de pantalla
710          DEC  C                     ;actualiza contador lineas
720          LD   A,C                   ;si no quedan lineas, fin del programa
730          CP   #00
740          JR   Z,fin
750          PUSH HL                    ;preserva puntero datos del fichero
760          LD   H,D
770          LD   L,E
780          CALL #BC26                 ;calcula direccion siguiente linea
790          LD   D,H
800          LD   E,L
810          POP  HL                    ;recupera puntero datos fichero
820          JR   contin                ;continua
830
840 ;                 rutina  PRIMER MARCADOR
850
860 mark1:   PUSH BC                    ;salva contadores de bytes y lineas
870          DEC  HL
880          LD   B,(HL)                ;carga no. de la linea igual
890          INC  B
900          PUSH HL                    ;preserva puntero datos fichero
910          LD   HL,#C000              ;busca linea a copiar
920 bucle:   CALL #BC26                 ;SCR NEXT LINE
930          DJNZ bucle
940          CALL #BC29                 ;SCR PREV LINE
950          LD   BC,#0050              ;copia la linea de (HL) a (DE)
960          LDIR
970          POP  HL                    ;recupera puntero fichero
980          DEC  HL                    ;siguiente dato del fichero
990          POP  BC                    ;recupera contadores bytes y lineas
1000         JR   otra
1010
1020 ;                SALIDA DEL PROGRAMA
1030
1040 fin:    LD   HL,(#9005)            ;carga OFFSET de pantalla
1050         CALL #BC05                 ;fija OFFSET de pantalla
1060         RET
```

Software & Hardware

"AMSTRAD 664" Serie II. CPM 2.2

RPA BASE DE DATOS — AD-BD087 — P.V.P. 9.500

Programa muy versatil, potente y de sencillo manejo. Con opciones de color, modificaciones, búsqueda, inserciones, etc. Dispone de una completísima salida por impresora.

RPA MASTER BASE — AD-MA079 — P.V.P. 12.500

Base de Datos realizada en PASCAL, arboles, búsqueda aleatoria.

RPA MULTICALC/CPM/PLUS — AD-HC024 — P.V.P. 12.500

Hoja de cálculo realizada en Pascal, le permite hacer cálculos matemáticos, así como gráficos. Es ideal para cálculos económicos y financieros.

RPA FACTURACION 664 — AD-FA070 — P.V.P. 12.500

Permite realizar facturas, almacenarlas y después modificarlas. Tratamiento de IVA. Totalizaciones generales y parciales. Genera recibos, albaranes y facturas. Salida por impresora muy completa.

ALMACEN-FACTURACION — AD-AF025 — P.V.P. 15.500

Control de entradas y salidas de inventarios con totalización de compras y ventas. Las facturas descuentan la mercancía del almacén automáticamente. Emite recibos, facturas y albaranes.
(Una unidad de disco).

RPA CLIENTES-PROVEEDORES-FACTURACION-MAILING — AD-CP071 — P.V.P. 15.500

Le permite tener a sus clientes y proveedores clasificados por sectores, sus totales facturados y su IVA. Totalización parcial y total. Permite hacer mailing y el control y seguimiento de la cartera de sus clientes.

PRA PROCETEXT — AD-PT061 — P.V.P. 9.500

Procesador de textos y mailing. Incluye márgenes, impresión, tabuladores, acentos, etc.

RPA AGENDA ROBOT — AD-AR032 — P.V.P. 12.500

Integrado por cuatro subprogramas: Agenda-Fichero de clientes, Dietario, Calculadora y Base de Datos.

CONTABILIDAD GENERAL II — AD-CG027 — P.V.P. 15.500

Contabilidad general española, según el plan contable. Incluye: Anotación de asientos, Modificación de apuntes, Asientos externos, Libro diario, Mayor, Balance, Fin de periodo, Fin de ejercicio. Con cálculos matemáticos realizados en ensamblador que garantizan alta precisión.

"AMSTRAD 6128" Serie Profesional 2000. CPM/Plus

RPA MULTI AGENDA ROBOT — AD-MR072 — P.V.P. 15.500

Realizada en Pascal.

RPA MULTIBASE 3 — AD-MB046 — P.V.P. 19.500

Permite hacer gráficos de curva, tarta e histogramas, lineal y en tres dimensiones.

RPA FACTURACION 6128 — AD-FA047 — P.V.P. 19.500

Incluye apuntes de contabilidad, formas de pago y diversas opciones de impresora: Factura, Albarán o Recibo. Con totalización parcial o general. Incluye IVA. (Opcionalmente dos unidades de disco).

RPA NOMINAS — AD-NO045 — P.V.P. 19.500

Comprende impresos TC1, TC2, apuntes de contabilidad e informes.

RPA GESTION DE EMPRESA — AD-GE030 — P.V.P. 65.000

Integración de los programas: Contabilidad General, Nóminas, Facturación, Clientes y Proveedores. (Dos unidades de disco).

RPA MASTER FILE — AD-MF048 — P.V.P. 19.500

Fichero maestro aleatoria con opción de impresora aleatoria. (Opcionalmente con dos unidades de disco).

RPA CONTABILIDAD GENERAL III — AD-CG062 — P.V.P. 19.500

Incluye masas patrimoniales.

RPA CONTABILIDAD GENERAL IV — AD-CG063 — P.V.P. 24.500

Incluye analítica y previsión de cobros.

RPA ALMACEN-FACTURACION — AD-AF069 — P.V.P. 19.500

Control de entradas y salidas de inventarios con totalización de compras, ventas y existencias. Emite recibos, facturas y albaranes. (Dos unidades de disco).

RPA CLIENTES-PROVEEDORES--FACTURACION-MAILING — AD-CP088 — P.V.P. 15.500

"AMSTRAD 6128" Educativos. CPM/Plus

RPA GEOGRAFIA DE ESPAÑA	AD-GS051	P.V.P. 9.500
RPA GEOGRAFIA UNIVERSAL	AD-GU073	P.V.P. 9.500
RPA CLIMATOLOGIA	AD-CL073	P.V.P. 9.500
RPA DEMOGRAFIA	AD-DE075	P.V.P. 9.500
RPA CUERPO HUMANO	AD-CH052	P.V.P. 9.500
RPA PLANETARIO	AD-PL076	P.V.P. 9.500
RPA TEOREMAS GEOMETRICOS	AD-TG077	P.V.P. 9.500

"AMSTRAD 6128" Sectores Profesionales. CPM/Plus

ARTICULO	REF.	P.V.P.
RPA MEDICO-CLINICO-FARMACOS	AD-CF028	15.500
RPA MEDICO-CLINICO-DENTISTA	AD-CE049	15.500
RPA CALCULO DE ESTRUCTURAS	AD-CE078	12.500
RPA VIDEO CLUB	AD-VC029	28.500

"AMSTRAD 8256" CPM/Plus

RPA ALMACEN-FACTURACION	AD-AF056	24.500
RPA CONTABILIDAD GENERAL 5	AD-GE055	24.500
RPA GESTION DE EMPRESA	AD-GE057	76.500
RPA MULTI AGENDA ROBOT	AD-MR085	19.500
RPA NOMINAS	AD-NO059	28.500
RPA MULTIBASE 3	AD-MB060	28.500
RPA FACTURACION 8256	AD-FA053	28.500
RPA MASTER FILE	AD-MF054	28.500
RPA VIDEO CLUB	AD-VC066	35.500
RPA GRAFICOS DE GESTION	AD-FG067	19.500
BASE DE DATOS "RPA ELEPHANT DATA BASE"	AD-ED080	24.500
RPA ADMINISTRACION DE FINCAS	AD-AD081	35.500
RPA PROJECT PLANNING	AD-PP082	19.500
RPA BRAINSTORM PLUS	AD-BS083	19.500
RPA CLIENTES-PROVEEDORES-FACT. MAILING	AD-CP084	24.500

Los programas para el Amstrad 8256 estarán disponibles en la última semana de enero. Sus características serán similares a los del Amstrd 6128, con mayor capacidad de datos y definición de pantalla. Todos ellos funcionarán con un sistema de protección (un Interfaz) denominado "Interproc".

Galileo, 25 - Entreplanta A - 28015-Madrid
Tels. 447 97 51 / 447 98 09

Los mejores programas del mes seleccionados por

AMSTRAD USER

PARTICIPE y GANE con nuestra revista uno de estos fabulosos premios

* 1 Impresora AMSTRAD DM-1
* 5 Lotes de 3 programas en cassette

- Para participar solamente deberá rellenar el cupón adjunto con los títulos de sus cinco programas favoritos en orden de preferencia y enviarlo a AMSTRAD USER.
- Todos los cupones recibidos antes del día 30 de cada mes entrarán en un sorteo.
- A los premiados les será notificado por carta certificada en su domicilio.



PREMIOS SORTEO AMSTRADIEZ DICIEMBRE 85

IMPRESORA AMSTRAD DMP-1
Oscar Gracia González
Avda. de Daroca, 29-2.°
28017 MADRID

LOTES DE 3 PROGRAMAS EN CASSETTE
Enrique Monrabal Company
C/ Alvarez Sotomayor, 27-6
46017 VALENCIA

José Luis Cano Cervera
C/ González Adalid, 7-6.° B
30001 MURCIA

Yolanda Pellejero Cezon
C/ Andrés Gurpide, 3, 7.° esc. 2
50008 ZARAGOZA

Francisco Javier López Serna
C/ Doctor Zamenhof, 2-4, 6.° 2.ª
08020 BARCELONA

Carlos Alfonso Rhodes de Diego
Pza .de la Iglesia ,9
Cabanillas del Campo
(GUADALAJARA)

AMSTRADIEZ MES: FEBRERO 1986

Programas:

1— ....

2— ....

3— ....

4— ....

5— ....

Nombre ....

Dirección ....

Localidad .... D.P. ....

Provincia ....

Profesión ....

Fotocopie o recorte y envíe este cupón a

AMSTRADIEZ

Avda. del Mediterráneo, 9
28007 Madrid

AMSTRADIEZ

MES: FEBRERO 1986

	PROGRAMA	MES PASADO	MES EN LISTA	ORDENADOR	CINTA	DISCO
1	ALIEN-8	3	5	TODOS	SI	
2	DECATHLON	2	5	CPC 464	SI	
3	KNIGHT LORE	4	5	TODOS	SI	
4	FIGHTER PILOT	1	5	TODOS	SI	SI
5	GREMLINS	5	5	TODOS	SI	
6	EXPLODING FIST	6	3	CPC 464	SI	
7	3-D VOICE CHESS	—	1	TODOS	SI	SI
8	PYJAMARAMA	8	3	TODOS	SI	SI
9	MANIC MINER	10	3	CPC 464	SI	
10	SIMULADOR DE VUELO 737	—	1	TODOS	SI	



VARIABLES DE NUESTRO PROGRAMA

La siguiente rutina es una útil herramienta para la depuración de programas, ya que lista las líneas en las que hemos utilizado una determinada variable BASIC.

Una (mediante MERGE) esta rutina con su programa y ejecútela con run 65350. También se puede utilizar como una subrutina cambiando los comandos END en las líneas 65360, 65410 y 65440 por RETURN.

La rutina pide el nombre de la variable y la línea por la que debe comenzar y acabar de inspeccionar el programa. Si no introducimos ningún valor, la búsqueda se efectúa sobre todo el programa. Para cada línea en que se utiliza la variable deseada obtenemos el número de línea y las veces que se emplea.

El programa realiza primero una serie de conversiones para adaptar el formato de entrada a la forma de manejar los nombres de variables del programa BASIC, y realiza a continuación una búsqueda por todo el programa. Al nombre de variable que le demos se le pone a uno el bit 7 del último carácter, para indicar el fin.

J. M.

```
65350 INPUT "Nombre de la Variable ";zz$
65360 IF zz$="" THEN END
65370 IF RIGHT$(zz$,1)="$" OR RIGHT$(zz$
,1)="!" OR RIGHT$(zz$,1)="%" THEN zz$=LE
FT$(zz$,LEN(zz$)-1):GOTO 65360
65380 zz$=UPPER$(zz$):zz1$=LOWER$(zz$):z
z$=LEFT$(zz$,LEN(zz$)-1)+CHR$(128 OR ASC
(RIGHT$(zz$,1))):zz1$=LEFT$(zz1$,LEN(zz1
$)-1)+CHR$(128 OR ASC(RIGHT$(zz1$,1)))
65390 zzinicio!=0:zzfin!=65535:INPUT "Li
nea de comienzo : ";zzinicio!:INPUT "Lin
ea de Finalizar: ";zzfin!:IF zzfin!=0 TH
EN zzfin!=65535
65400 zz!=&170
65410 zz1!=PEEK(zz!)+256*PEEK(zz!+1):IF
zz1!=0 THEN PRINT "Encontrado el fin del
 programa.":END
65420 zz2!=PEEK(zz!+2)+256*PEEK(zz!+3)
65430 IF zz2!<zzinicio! THEN zz!=zz!+zz1
!:GOTO 65410
65440 IF zz2!>zzfin! THEN END
65450 zz0!=0:zz3!=zz!+4
65460 IF zz3!>=zz!+zz1! THEN 65490
65470 IF CHR$(PEEK(zz3!))=LEFT$(zz1$,1)
OR CHR$(PEEK(zz3!))=LEFT$(zz$,1) THEN GO
SUB 65510
65480 zz3!=zz3!+1:GOTO 65460
65490 IF zz0!<>0 THEN PRINT "En la linea
 ";zz2!;" - ";zz0!;" veces"
65500 zz!=zz!+zz1!:GOTO 65410
65510 zz5!=zz3!:FOR zz4!=1 TO LEN(zz1$)
65520 IF NOT (MID$(zz1$,zz4!,1)=CHR$(PEE
K(zz5!)) OR MID$(zz$,zz4!,1)=CHR$(PEEK(z
z5!))) THEN RETURN
65530 zz5!=zz5!+1:NEXT zz4!:zz0!=zz0!+1:
zz3!=zz3!+LEN(zz1$):RETURN
```

CURVAS EN TRES DIMENSIONES

Este programa de 14 líneas produce algunas de las mejores figuras en tres dimensiones.

El único inconveniente es que emplea varios minutos en dibujar las figuras más elaboradas pero merece la pena esperar. Podemos obtener efectos interesantes cambiando la línea 60 por:

60 Q = (R-1)*COS(6-R)

Y pruebe también:

60 Q = (R-1)*SIN(– 16*R)

En la línea 10 están definidas las variables Xs y YS que controlan la definición del dibujo; ponga simplemente un número menor para obtener puntos más juntos. XS = 1 y YS = 1 nos da la mayor definición pero emplea consecuentemente mayor tiempo en realizar el dibujo. Use XS = 6 y YS = 6 para obtener una rápida idea de qué va a aparecer con cada cambio.

J. M.

```
10 MODE 2:ORIGIN 320,200:xs=4:ys=4
20 a=320:b=a*a:c=200
30 FOR x=a TO 0 STEP -xs:s=x*x:p=SQR(b-s)
40 FOR i=-p TO p STEP 6*ys
50 r=SQR(s+i*i)/a
60 q=(r-1)*SIN(16-r)
70 y=i/3+q*c
80 IF i=-p THEN m=y:GOTO 110
90 IF y>m THEN m=y:GOTO 120
100 IF y>=n THEN GOTO 130
110 n=y
120 PLOT -x,y:PLOT x,y
130 NEXT:NEXT
140 END
```

MANEJO DIRECTO DEL *CHIP* DE SONIDO

Muchos de los usuarios de Amstrad ya sabrán que el controlador de sonido que incorpora, el AY-3-8912, es un chip muy popular, y que se utiliza en muchos ordenadores. Sin embargo, el manejo a través del BASIC y el operativo de su sonido se hace por una codificación intermedia que dificulta el acceso directo a los registros internos.

Aunque su programación directa puede llegar a ser tediosa, nos proporciona la ventaja de poder estirar al límite sus posibilidades. El programa que os presentamos a continuación crea el comando (RSX): SONIDO. Este comando tiene dos parámetros: registro y valor. Su cometido es muy sencillo, pero no será fácil de usar sin saber el cometido de estos registros. Los tres ejemplos que lo acompañan ayudarán a conseguir tres bonitos efectos sonoros siempre, claro está, que antes se haya cargado el primer listado para definir el comando.

O. L.

```
10 REM
20 REM COMANDO :SONIDO,REGISTRO,VALOR
30 REM
40 MEMORY 33999
50 FOR a=34000 TO 34036
60 READ x:POKE a,x
70 NEXT A
80 CALL 34000:NEW
90 DATA 33,228,132,1,218,132,205,209,188
,201,222,132,24,12,83,79
100 DATA 78,73,68,207,252,166,218,132,0,
0,221,126,0,79,221,126,2,205,52,189,201


10 REM EXPLOSION
20 :SONIDO,7,&X10110111
30 :SONIDO,8,31::SONIDO,6,120
40 :SONIDO,11,4::SONIDO,12,90
50 :SONIDO,13,0

10 REM DISPAROS
20 :SONIDO,7,&X10110111
30 :SONIDO,8,31::SONIDO,6,120
40 :SONIDO,11,24::SONIDO,12,10
50 :SONIDO,13,8

10 REM SILBIDOS
20 :SONIDO,7,&X10111110
30 :SONIDO,8,15
40 FOR A=255 TO 1 STEP -1::SONIDO,0,A:NE
XT A
50 FOR A=1 TO 255::SONIDO,0,A:NEXT A::SO
NIDO,7,255
```

EL AMSTRAD ARCOIRIS

Se puede crear un carácter multicolor en el AMSTRAD CPC 464 usando el modo transparente (PRINT CHR $(22); CHR $(1). Fijando este modo en el ordenador e imprimiendo diferentes caracteres coloreados uno sobre otro, podremos crear caracteres repletos de colorido.

```
5 MODE 0
6 SYMBOL AFTER 190
10 SYMBOL 200,16,56,16,0,0,16,24
20 SYMBOL 201,0,0,0,40,108,0,0,0
30 SYMBOL 202,0,0,0,16,16,40,16,0
40 FOR x=1 TO 20
50 PRINT CHR$(22);CHR$(1)
60 CLS
70 PEN 1
80 LOCATE x,10
90 PRINT CHR$(200)
100 PEN 2
110 LOCATE x,10
120 PRINT CHR$(201)
130 PEN 3
140 LOCATE x,10
150 PRINT CHR$(202)
160 PRINT CHR$(22);CHR$(0)
170 NEXT x
```

FORMATO DE IMPRESION EN EL AMSTRAD

Algunos lectores nos han escrito quejándose del formato de impresión del AMSTRAD y señalando que si una frase va a sobrepasar el límite derecho de la pantalla, se genera un <RETORNO DE CARRO> y la frase «salta» a la línea siguiente:

```
10 MODE 1
20 A $ µstring $ (22,«X»)
30 PRINT A $; A $
```

Este ejemplo nos dará al ejecutarlo dos líneas separadas de A $. Sin embargo, esto no es un error del BASIC, sino una característica intencional. Para evitar esto, en lugar de PRINT utilice USING«&»; por ejemplo, cambie la línea 30 por:

```
30 PRINT USING «&»; A $; A $
```

y le imprimirá A $ dos veces en la misma línea.

Ocurre exactamente igual con el comando LOCATE:

```
30 LOCATE 30,1: PRINT A $
```

y le escribirá A $ en la siguiente línea, mientras que

```
30 LOCATE 30,1: PRINT USING «&»; A $
```

le producirá el resultado esperado.

A continuación ofrecemos una forma de utilizar los códigos ASCII menores de 32 directamente desde el teclado como caracteres. Teclee PRINT» y pulse [CTRL] y una letra y luego [ENTER]. Pruebe los siguientes ejemplos:

PRINT «[CTRL] G» - produce un pitido.
PRINT«[CTRL] DO» - nos da el modo cero.
PRINT «[CTRL] O3» - nos da color rojo (si estamos en modo cero).

Puede intentar poner esto en un programa de esta forma:

```
10 MODE 0
20 PRINT «[CTRL] O1aaaaa, O2bbbbb, O3ccccc»
```

Este ejemplo producirá una línea multicolor de amarillo, cyan y rojo.

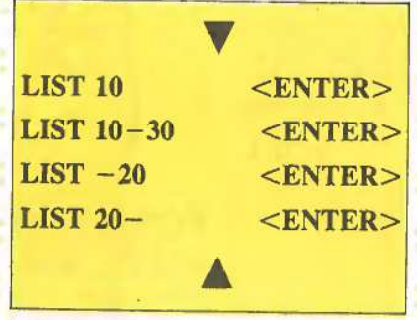
curso de programación

LENGUAJE BASIC



El mes pasado acabábamos con un pequeño programa. En este segundo capítulo prepararemos un nuevo programa con el que, por un lado, aprenderemos nuevas instrucciones y por otro entraremos en contacto con algunos conceptos de programación. Algo que se mencionó en la primera parte fue que los programas se almacenaban en memoria. La memoria no sólo es el lugar donde se guardan los programas; también es posible «depositar» listas de nombres o números, incluso el propio sistema tiene almacenadas gran cantidad de útiles en ella. Todo ello lo estudiaremos en detalle más adelante. Por el momento, nos interesa saber cómo observar el programa que tenemos en memoria. Para ello, indicamos al AMSTRAD que nos proporcione un LISTado de las instrucciones que previamente hayamos introducido. Una vez tecleado un programa, por ejemplo el del capítulo anterior, teclearemos LIST y pulsaremos <ENTER>. Con ello visualizaremos el conjunto de instrucciones tecleadas ordenadas por su número.

Si sólo deseamos ver parte de esas instrucciones, el comando LIST permite seleccionar la parte buscada del listado del programa. Basta indicarlo con un número a continuación. Por ejemplo:



```
LIST 10        <ENTER>
LIST 10-30     <ENTER>
LIST -20       <ENTER>
LIST 20-       <ENTER>
```

El primer LIST solicita que se liste únicamente la sentencia número 10, el segundo listará las sentencias entre la 10 y la 30 (ambas incluidas), la tercera saca un listado hasta la sentencia número 20 y, por último, la cuarta lista todas las sentencias a partir de la número 20 hasta el final. La posibilidad de efectuar listados de programas permite, aparte de comprobar qué se ha escrito, repasar los errores de conjunto y verificar la corrección de la escritura.

Supongamos que desea eliminar de la memoria el programa tecleado, es decir, aquello que NOSOTROS hemos ido almacenando. Si tecleamos NEW y después pulsamos <ENTER> todo cuanto hubiera almacenado en memoria desaparecerá. Así, en el caso que vimos no ocurriría absolutamente nada al intentar ejecutar el programa. No conviene hacer pruebas con el NEW después de haberse pasado un par de horas tecleando (los nervios de mucha gente no lo resisten).

Continuemos con la programación. Un programa como el siguiente:



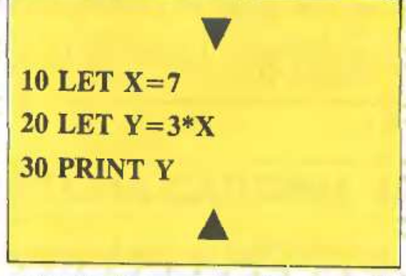
```
10 CLS
20 PRINT "23"
30 PRINT
40 PRINT "NUMERO"
```

Cada vez que se ejecute escribe el número 23. Pero..., ¿y si deseáramos imprimir en cada ocasión un número diferente? La solución es disponer de una especie de caja o recipiente donde depositar el número y que el programa imprimiera lo que encontrase en la caja en las sucesivas ejecuciones. Eso es una variable. No es más que un lugar de la memoria del ordenador cuyo contenido no tiene por qué ser siempre el mismo, sino que varía a lo largo de la ejecución de un programa. Para trabajar con las variables ocurre al igual que con los buzones de correos: es necesario darles un nombre que permita su identificación diferenciándolas de las demás.

Los nombres de las variables se forman con letras y números, teniendo en cuenta que el primer carácter debe ser una letra. Suele ser recomendable que los nombres de las variables hagan alusión a su significado real. Por ejemplo, si una variable contiene el precio del pan, un nombre significativo sería: precpan.

No conviene tampoco que los nombres sean demasiado largos ya que si son variables que deben teclearse en múltiples ocasiones, la introducción de los programas se vuelve farragosa y aburrida.

Escribamos el siguiente programa:



```
10 LET X=7
20 LET Y=3*X
30 PRINT Y
```

En él observamos varias cosas. En primer lugar aparece una nueva instrucción: LET. La sentencia que incluye LET se denomina sentencia de asignación. Es usual traducir LET X=7 por «haz X igual a 7». Ello equivale a indicar que se almacene en la «caja» con el nombre X el número 7. Dicho de manera más propia: asigna a la variable X el valor 7. Así, mientras no aparezca

una nueva asignación para esta variable, el ordenador interpreta que el valor de X es 7. La sentencia 20 asigna a la variable Y el valor 3 * 7 (es decir, 21). La sentencia 30 indica al ordenador que escriba en la pantalla el valor de la variable Y, que no es lo mismo que la letra «y». En las sentencias de asignación, la variable que cambia de valor aparece situada a la izquierda del signo "=", mientras que a la derecha puede aparecer un número, una expresión a calcular o simplemente otra variable.

Es importante no identificar el signo "=" de las sentencias de asignación con el significado del signo matemático igual. Así, por ejemplo, las sentencias:



```
10 LET X=7
20 LET X=X+1
30 PRINT X
```

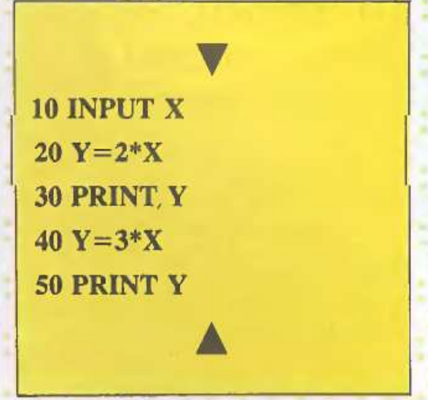
tienen un significado muy claro en BASIC:

10 "haz que X valga 7"
20 "haz que X valga el valor que tenía X más uno" (hacer que X valga 7 + 1, es decir, 8)
30 "escribe el valor de X" (escribe el número 8)

mientras que matemáticamente la ecuación X=X + 1 carece de sentido. Así pues, es importante concebir una variable como el nombre asignado a una zona de memoria donde se almacena su valor. Por supuesto, esta zona de memoria física es única, de forma que el ordenador una vez ejecutada la instrucción de asignación (por ejemplo X=X + 1) se almacena el número 8 olvidando (en realidad escribiendo encima) el valor anterior. En la mayor parte de los ordenadores, en concreto en el AMSTRAD, no es necesario teclear la palabra clave LET en las sentencias de asignación. Por ello, a partir de ahora omitiremos su escritura.

Resulta interesante observar cómo en el último programa sería de gran utilidad no tener que emplear una sentencia de asignación para conseguir que la variable X tenga el valor 7, sino que el ordenador nos permitiera, durante la ejecución del programa, indicar cuál es ese valor.

Esa es la misión de la instrucción INPUT. Cuando en un programa aparece esta instrucción, el ordenador se detiene esperando que se le comunique el valor que debe asignar a la variable. Así la ejecución del siguiente programa:



```
10 INPUT X
20 Y=2*X
30 PRINT Y
40 Y=3*X
50 PRINT Y
```

sería:

10 Aparece en la pantalla un signo, normalmente "?" y el cursor. Nosotros tecleamos un número y pulsamos la tecla <ENTER>. Con ello el AMSTRAD asigna a la variable el número tecleado.

20 El ordenador asigna a la variable Y el valor de la expresión 2*X, es decir, el doble del valor introducido en la sentencia anterior.

30 Aparece en pantalla el valor de Y.

40 Se la asigna a Y el valor de 3*X. Téngase en cuenta que desaparece el valor que contenía Y.

50 se escribe el nuevo valor de Y.

Tratemos de escribir ahora un programa con un significado claro. Tomaremos un ejemplo de la vida cotidiana. Vamos a realizar el cálculo de gastos de una noche de fiesta. Primero calcularemos el precio de la cena multiplicando el precio del menú por el número de asistentes y después el coste de la sala de fiestas por un procedimiento similar.

El programa quedaría así:



```
10 INPUT PRECMENU
20 INPUT NUMASIST
30 COSTECENA = PRECMENU * NUMASIST
40 INPUT PRECSALA
50 COSTESALA = PRECSALA * NUMASIST
```

y con ello el coste total resulta:

```
60 COSTOTAL = COSTECENA + COSTESALA
70 PRINT COSTOTAL
```

Como se observa, sería de interés poder indicar cuál es el objetivo del programa. La ventaja es que una persona que no haya escrito este programa podrá identificar su cometido y sus partes sin dificultades. La instrucción REM se usa con esta función. Una sentencia encabezada por REM especifica al ordenador que no debe interpretar esa sentencia. Es decir, esa línea figura en el programa como un simple comentario. Al pasar por ese número de línea la ejecución del programa la ignorará. En nuestro ejemplo podríamos haber puesto:

```
5 REM PROGRAMA PARA CALCULAR GASTOS DE FIESTAS
```

e incluso:

```
55 REM CALCULO DEL TOTAL
```

Además, el programa ejemplo tiene otro defecto importante. Durante su ejecución (comprueba el funcionamiento) la pantalla no nos ayuda en la interpretación de los datos. Si uno ejecuta el programa y ve aparecer el signo "?" no sabrá qué debe introducir: el precio de la cena o el número de comensales. Para poder utilizar un programa así es imprescindible recordar el orden en que habían de ser introducidos los datos. En programas de mayor complejidad llega un momento en que no es posible acordarse de qué dato está solicitando el ordenador. La forma de evitar tal desorganización es simple. Un modo de hacerlo puede ser intercalar (he aquí la ventaja de haber numerado de 10 en 10) entre cada solicitud de entrada de un dato una instrucción PRINT que indique el significado.

En el ejemplo sería:

```
5 PRINT "INTRODUZCA EL PRECIO DEL MENU POR PERSONA"
15 PRINT "INTRODUZCA EL NUMERO DE ASISTENTES"
35 PRINT "INTRODUZCA EL PRECIO DE LA ENTRADA A LA FIESTA"
65 PRINT "EL COSTE TOTAL DE LA NOCHE ES DE:"
```

El programa que resulta es más fácil de comprender por quien lo ejecuta. De todas formas, es posible mejorar su estructura:

La instrucción PRINT permite la escritura de textos y variables. También permite escribir varios textos y variables separando unos de otros mediante los signos "," y ";". Cuando se utiliza el punto y coma (;) la siguiente escritura se realiza justo a continuación del último símbolo escrito. Cuando se escribe un número éste aparece con un espacio a cada lado. Así el programa:

```
100 a=7
110 b=15
120 PRINT "numeros" ; a;b;a; "fin"
```

resulta (al ejecutarlo con RUN 100):

```
números 7 15 7 fin
```

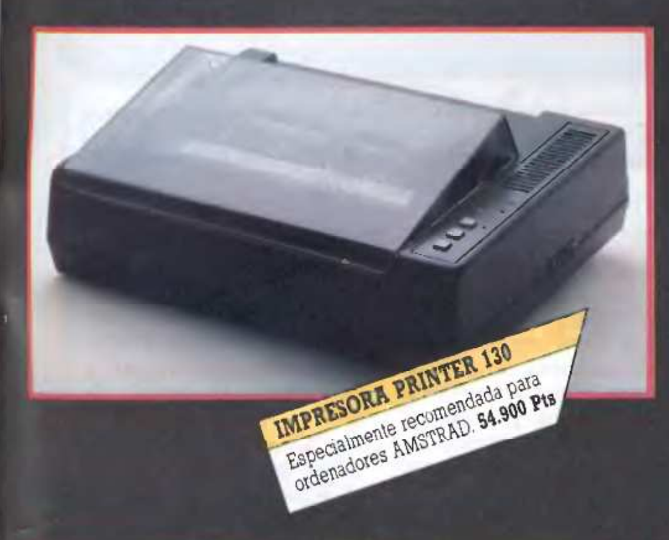
Las distintas cadenas y variables se pueden separar también por el signo ",". La línea de escritura del ordenador se divide en el AMSTRAD en tres sectores. El empleo de la coma (,) indica que la próxima escritura se realizará en el siguiente sector.

Como ejercicio para este mes vamos a dejar el repaso de todo lo visto y la realización de un programa que, además de calcular el coste de esa noche, dé una salida en pantalla de este estilo:

```
COSTE DEL MENU
1230
COSTE DE LA SALA
1000
COSTE TOTAL = 6690 PTS
NUMERO DE ASISTENTES
3
NUMERO DE ASISTENTES
3
COSTE DE LA CENA
3690
COSTE DE FIESTA
3000
```

Luis Alvarez

AMPLIA LAS POSIBILIDADES DE TU AMSTRAD



IMPRESORA PRINTER 130
Especialmente recomendada para ordenadores AMSTRAD. **54.900 Pts**



JOYSTICKS
Los famosos SVI de la serie Quickshot. Desde **1.600 Pts**

LAPIZ OPTICO
Diseña gráficos y menús de comunicación en la pantalla a color. Incluye software. **4.500 Pts**



INTERFACE SERIE RS 232 C
Para conectar con modems, impresoras serie u otros ordenadores. **11.750 Pts.**



UNIDAD DE DISCO
Incluye Sistema Operativo CP/M y lenguaje LOGO.
(con controlador). **45.500 Pts**
(sin controlador). **39.500 Pts**

SINTETIZADOR DE VOZ
Emula la voz humana. Incluye dos altavoces y el software. **9.000 Pts**

MODULADOR TV COLOR
Para utilizar el TV como pantalla a todo color. **9.000 Pts** (CPC 464), **9.450 Pts** (CPC 664 y 6128)

AMSTRAD ESPAÑA
GRUPO INDESCOMP

Avda. del Mediterráneo, 9. Tels. 433 45 48 - 433 48 76. 28007 MADRID
Delegación Cataluña: Tarragona, 110 - Tel. 325 10 58. 08015 BARCELONA



Libro: Inteligencia artificial: conceptos y programas.
Autor: Tim Hartnell

Editorial: Anaya Multimedia
Páginas: 270

Todos hemos visto en alguna película máquinas-robots-ordenadores pensantes, capaces de tomar decisiones inteligentes, o hemos leído su descripción en más de una novela de ciencia-ficción. Sabemos incluso que existen realmente ordenadores muy potentes, preparados para ayudar en diagnósticos médicos, control de procesos industriales, navegación, etc., si bien sólo al alcance de científicos escogidos, que nuestra imaginación representa enfundados en su típica bata blanca. Ahora bien, pocos se atreverían a imaginar la «inteligencia artificial doméstica».

Sin embargo, en un libro verdaderamente ameno, Tim Hartnell nos demuestra que es posible; y no sólo eso, sino que además nos enseña a que seamos nosotros mismos los que desarrollemos nuestros propios programas «inteligentes». Toda una aventura que llegará a buen término si nos dejamos guiar por la mano firme y segura de esta excelente obra.

Así, entre otros, se nos presentan programas como SILOGISMOS, capaz como su nombre indica de seguir líneas deductivas totalmente al estilo de los antiguos filósofos griegos. O el caso de DOCTOR, una versión BASIC del famoso ELIZA, que probablemente pueda ayudarle a resolver esos pequeños problemas anímicos o psíquicos. Incluso podremos encontrar un programa que jugará con nosotros de forma incansable a un juego semejante a las damas, con la peculiaridad de que el ordenador irá aprendiendo a medida que transcurran más y más partidas. Dentro de la línea de programas de juego, podremos también disputar grandes partidas de 3 en Raya. Además se nos ofrecen programas tales como un generador de poemas, o un sistema experto para elegir microprocesadores, o un programa que, siguiendo nuestras órdenes, mueve objetos en un mundo bidimensional, e incluso un eficiente traductor inglés/español.

Posiblemente usted, lector, no conozca el tema, o tal vez piense que no le interesa. Pero seguro que al leerlo cambiará de opinión, ya que la presentación de los diversos problemas planteados —razonamiento humano, razonamiento simulado, programas de juegos inteligentes, sistemas expertos, programas que aprenden de sus errores, programas traductores —está muy bien estructurada, analizando los temas primeramente desde un punto de vista teórico, para pasar a continuación al estudio fragmentado del correspondiente programa-ejemplo propuesto por el autor. Dichos ejemplos están realizados en un BASIC estandard, facilmente adaptable a la mayoría de los ordenadores domésticos que se encuentran en el mercado. Incluso se ofrece al final un apéndice con los listados adaptados al ZX-Spectrum.

La traducción es excelente, muy por encima de lo que es habitual en los libros sobre ordenadores domésticos, y los programas son igualmente buenos, si bien el autor no cesa de invitarnos a que investiguemos la forma de mejorarlos e incluso a que desarrollemos nuestros propios programas. Para ese fin incluye un apéndice con consejos muy útiles sobre el método a seguir en el desarrollo de un programa.

Otro punto a destacar es la constante información bibliográfica, de la cual encontramos una recopilación en otro apéndice. Sin duda un elemento interesante para aquellos que deseen seguir investigando y profundizando en este tema. Hemos de descubrirnos ante el notable esfuerzo realizado por el autor para documentarse adecuadamente y poder incluso ofrecer a los lectores opiniones contrastadas en temas tan conflictivos como la frontera entre razonamiento e inteligencia, o el problema de hasta dónde se debe permitir llegar a las máquinas en cuestiones de inteligencia.

Tim Hartnell
Inteligencia Artificial:
conceptos y programas
ZX Spectrum - Spectrum + - VIC 20 - Commodore 64 - Apple IIe, IIc - Dragon 32, 64 - IBM pc - Oric - MSX

Guía de especialistas de AMSTRAD USER

Libro: Ordeno y Aprendo con AMSTRAD para EGB
Autor: J. L. Carralero
Editorial: INED, S. A.
Páginas: 193

«Ordeno y Aprendo» es algo más que un curso de iniciación al BASIC. Se trata de un libro capaz de enseñar a cualquiera el camino exacto para realizar un programa con éxito, incluso a los más pequeños. La estrategia pedagógica es excelente, y sigue un modelo de aprendizaje configurado en dos apartados.

En primer lugar, se aborda el conocimiento necesario para la asimilación de los conceptos básicos y su aplicación en las técnicas de trabajo, para continuar con el área práctica, enfocada al dominio y manejo del ordenador mediante la utilización del lenguaje BASIC de programación. Al final de cada capítulo se incluye una serie de preguntas para reafirmar los conocimientos adquiridos.



El libro está graciosamente ilustrado con tortugas semejantes a las del LOGO, que se encargan de representar en imágenes lo que el texto nos explica con palabras.

Al principio del libro se nos ofrece un resumen de la historia de la informática, así como una primera aproximación al ordenador, el hardware y el software. A continuación se analiza el ordenador más en profundidad, especialmente su estructura interna. Simultáneamente se empieza con los comandos BASIC.

Ahora nos encontramos con los sistemas de numeración, cambios de base, las variables, las constantes, el tratamiento de la información. Y también, cómo no, los organigramas, las bifurcaciones y las variables de control.

Por último, se analiza el uso de los bucles, las instrucciones FOR... NEXT, las instrucciones READ y DATA, el comando EDIT, y los poderosos AUTO y RENUM. Y como colofón, se explica la utilización y dimensionado de tablas.

Libro: «Código máquina para principiantes con AMSTRAD
Autor: Steve Kramer
Editorial: Indescomp
Páginas: 175

Este libro va dirigido al principiante que desea aprender a programar en código máquina en el AMSTRAD CPC464. En primer lugar se apuntan unas breves nociones elementales sobre numeración binaria y hexadecimal, los códigos ASCII y una primera aproximación al lenguaje ensamblador.

A continuación se explica lo que es un diagrama de flujo y los símbolos más utilizados.

A partir de este momento nos introducimos en el mundo fascinante del Z80. Empezamos con los registros y las instrucciones que permiten mover información de un registro a otro. Se analiza cada instrucción y, en los casos en que es posible, se muestra el paralelismo existente con el lenguaje BASIC.

Pasamos ahora al Contador de Programa (registro PC), y su manejo por medio de las instrucciones CALL y JP. También se explica en esta sección el manejo del STACK realizado por las instrucciones CALL.

Abordamos después las instrucciones aritméticas del Z80, tales como IND, DEC, SUB, SBC, ADD, ADC, así como la explicación de conceptos como el de acarreo. Inmediatamente después se explica el registro de indicadores (flags) y su relación con las instrucciones anteriores y con otras como CP y los saltos condicionales, tanto relativos como absolutos. Se incluyen también aquí las instrucciones CCF y SCF.

Continuamos con las operaciones lógicas (AND, OR, XOR, CPL y NEG), y con un análisis más detallado de la utilización del STACK mediante las instrucciones PUSH y POP.



Llegados a este punto nos sumergimos de cabeza en el nutrido grupo de instrucciones de rotación: RL, RLA, RLC, RLCA, RLD, SLA, RR, RRA, RRC, RRCA, RRD, SRA, SRL. Nos habla también de sus efectos en los flags y su aplicación para operaciones en matemática binaria.

Por último, las instrucciones de carga LDD, LDDR, CPD, CPDR, LDI, LDIR, CPI, CPIR, y las instrucciones de entrada y salida IN y OUT.

Un último apartado se refiere a las instrucciones menos frecuentes, como las de interrupciones y entrada-salida de cadena. Habla también en este apartado de los restarts y los modos de direccionamiento.

El final del libro está constituido por una breve referencia al modo de utilizar las rutinas del Sistema Operativo (tal vez demasiado breve) y una serie de apéndices (tablas de instrucciones del Z80, conversión hexadecimal-decimal, mapa de pantalla del AMSTRAD y direcciones de algunas rutinas del Sistema Operativo).

Guía de especialistas de AMSTRAD USER

CANARIAS



REMSHOP

ORDENADORES PERSONALES
Y DE GESTION EMPRESARIAL

ESPECIALISTAS EN PROGRAMAS EDUCATIVOS Y DE GESTION

GRAL. MAS DE GAMINDE, 45
Tel. 23 02 90
LAS PALMAS DE GRAN CANARIA

CANARIAS

"Equintesa"
INFORMATICA

ESPECIALISTAS
EN SISTEMAS LLAVE EN MANO
CON AMSTRAD

San Sebastián, 74 - Ofic. 31
Tels (922) 21 06 04 - 22 46 65 (Contest.)
38005 SANTA CRUZ DE TENERIFE

EL FERROL

MASTER COMPUTER

DISTRIBUIDOR OFICIAL AUTORIZADO

C/ Magdalena, 118
Tel. (981) 35 49 83
EL FERROL

JAEN



OFIMATICA

Especialistas en programas
y periféricos para AMSTRAD

PROFESIONALES
A SU SERVICIO

LINARES	JAEN
Alfonso X, 34	Pasaje Maza, 7
Tel. 69 80 52	Tel. 25 01 44

LOGROÑO

eguizábal

INFORMATICA
ELECTRONICA
TELECOMUNICACIONES

DRS. CASTROVIEJO, 34
Tel. (941) 23 12 82
26003 LOGROÑO

MADRID

MASTER COMPUTER

Centro Comercial, local 15
Ciudad SANTO DOMINGO
Carretera de Burgos, Km. 26
Tel.: 622 12 89 Algete Madrid.

Centro Comercial "EL BOULEBARD"
La Moraleja
Alcobendas Madrid
Tel.: 654 16 12

ABIERTO DOMINGOS DE 10 a 3 H.

MADRID

MASTER COMPUTER

CENTRO COMERCIAL
LOCAL 15

Tel. 622 12 89
CIUDAD SANTO DOMINGO
ALGETE (MADRID)

ABIERTOS LOS DOMINGOS

MADRID

KC INFORMATICA, S.A.

LO QUE TU NECESITAS
Y A UN BUEN PRECIO

ORDENADOR
AMSTRAD

DISTRIBUIDORES DE PRODUCTOS
INFORMATICOS

Hermosilla, 75 - 1.° Ofic. 14
Tel. (91) 276 43 94 - 28001 MADRID

MADRID



PASEO CASTELLANA, 126
28046 MADRID

PUERTO RICO, 21-23
Tel. 250 74 02 - 04
28016 MADRID

* Distribuidor oficial autorizado



CORREO

Estimados amigos:

Soy un suscriptor de esta maravillosa revista y desearía que me aclarasen un problema que no permite la ejecución de un programa que presentaron en su revista. Me refiero al conocido juego de la rana del que ofrecieron un listado en el número 2 de su revista perteneciente al mes de noviembre de 1985.

Después de haber copiado el programa, me dispuse a ejecutarlo mediante la orden RUN y el ordenador emitió el siguiente mensaje de error:

UNKNOWN COMMAND IN 470

El listado de la línea 470 es el siguiente:

470 |RSCR1,5,6,11,12,9,10,9,10: |LSCR1,7,8: |RSCR2,15,16,17,18: |LSCR2,21,22,19,20,19,20.

El mencionado error viene a decir que el BASIC no encuentra un comando externo que corresponde a lo escrito trás el símbolo '|' (según la definición del libro del usuario).

Les advierto que el juego sale normalmente (todo funciona correctamente) hasta que pulso la tecla [J] (para Jugar). Entonces aparecen los dibujos generales en la pantalla e, inmediatamente después, el mencionado error que interrumpe la ejecución. También deseo advertirles que poseo un CPC-664 y el problema puede venir dado por la incompatibilidad del programa.

Así pues, me gustaría que me dieran una solución precisa al problema pues me gustaría tener el emocionante juego a mi alcance.

Quisiera también que intentaran solucionar lo que creo que es un fallo del ordenador. Este problema viene dado por la aparición del mensaje SINTAX ERROR IN (línea x) cada vez que el ordenador encuentra en la línea x la orden DEFINIT a-z. Cuando ocurre esto cambio la orden DEFINIT por la de DEFREAL con lo cual el programa sigue su curso aunque querría saber si esto perjudica o no su ejecución.

Con las gracias anticipadas se despide un lector y seguidor de su revista que les felicita por su trabajo.

Muy cordialmente,

Pere Juanola González
Roses (Gerona)

En lo referente al programa 'la rana', el problema es el siguiente:

Los comandos RSX (o lo que es lo mismo, los comandos precedidos por el símbolo '|' no son ni más ni menos que rutinas en código máquina, a las que se da un nombre mediante una tabla de nombres y, utilizando una rutina del sistema operativo, se informa al BASIC de que dichos comandos y dicha tabla de nombres, se produce el citado mensaje de error, ya que no encuentra el nombre que busca.

Efectivamente, según el texto de su carta, la línea 470 está bien escrita, por lo que el fallo habrá que buscarlo en la tabla de nombres. ¿Y dónde está esta famosa tabla?

Bien, analicemos la situación. Tanto las rutinas en máquina (que constituyen los comandos RSX) como la tabla de saltos y la llamada a la rutina del sistema operativo para inicializar dicha tabla se encuentran en los DATAS del programa en BASIC. Así, probablemente el error se encuentre en alguno de estos datos, y más concretamente en los referentes a la tabla de saltos. Por tanto, sólo resta aconsejarle que revise cuidadosamente el contenido de las líneas 3260 a 3290, ambas incluidas, ya que probablemente esté ahí el fallo.

En cuanto a su segunda pregunta, la solución es muy simple: basta con sustituir el nombre incorrecto del comando (DEFINIT) por el nombre correcto (DEFINT) y todo resuelto.

Somos dos lectores de su revista, que por cierto nos gusta mucho, y referente al artículo que publicaron en el número 3 de su revisrta sobre el juego «Knight Lore» y los posibles cambios que en él se pueden hacer (vidas infinitas, tiempo infinito, cambio de objetos, etc.) desearíamos que nos lo enviasen o lo publicasen en la sección de correo con más claridad.

Alberto y Antonio
Valencia

En la página 30 del número 3 de nuestra revista aparecen dos listados en BASIC. El primero de ellos permite, siguiendo las instrucciones del texto del artículo, cambiar el número de objetos necesarios para completar el juego, el número de vidas y el tiempo disponible. El segundo permite cambiar los objetos que aparecen en la habitación cero y, leyendo atentamente el texto, verán que no es difícil modificarlo para cambiar otra habitación.

Si su pregunta se refiere a si podemos publicar el ***Knight Lore*** mismo, no es posible, ya que se trata de un juego comercial, protegido por derechos de autor y comercializado en España. El juego en sí debe ser comprado en alguna tienda especializada. Si pudiéramos lo publicaríamos, pero a la gente de Ultimate no le iba a sentar muy bien. Lo que pretende el artículo es facilitar el recorrido del juego y hacer que llegar al final sea accesible a los mortales.

Guía de especialistas de AMSTRAD USER

MADRID

BOUTIQUE AMSTRAD

INFORMATICA PERSONAL

CLARA DEL REY, 58 TELEFONO 415 15 46

TODO, ABSOLUTAMENTE TODO PARA SU AMSTRAD

MADRID

BAZAR TETUAN

ESPECIALISTAS EN AMSTRAD

Arenal, 9 Tel. 265 68 55

ORENSE



ALMACENES MENDEZ

Distribuidor Oficial de:

AMSTRAD ESPAÑA

Venga a visitarnos

Capitán Cortés, 17

Tel. (988) 22 86 07 32004 ORENSE

PONTEVEDRA



GABINETE DE ECONOMISTAS AUDITORES DE EMPRESA, S.A.

Benito Corbal, 17 - 1ª Dcha

Tel. 84 69 12 - PONTEVEDRA

SAN SEBASTIAN



DESCUENTOS

OFERTAS ESPECIALES

DISTRIBUIDOR OFICIAL AUTORIZADO

Claudio Delgado, 4 - Tel. 27 38 11

20001 SAN SEBASTIAN

VALLADOLID



JUAN DE JUNI, 3

Tel. 33 40 00

47006 Valladolid

* Distribuidor oficial autorizado

VALENCIA

omicron

DISTRIBUIDORES PARA CENTROS DE ENSEÑANZA DE LA COMUNIDAD VALENCIANA

C/Serpis (Junto Plaza Xúquer)

Tel. 361 05 08

Maestro Palau, 12

Tel. 331 53 27 VALENCIA

ZARAGOZA

EN ZARAGOZA

Encontrarás:

Todos los programas originales en cassette y diskette

Equipos de AMSTRAD — Periféricos

Libros — Discos vírgenes

De venta en:

Runa Distribuidor Oficial AMSTRAD

Duquesa Villahermosa, 3

50010 Zaragoza | Tel (976) 35 09 48

ENVIOS CONTRA REEMBOLSO

ZARAGOZA

Chips & Tips

LEON XIII, 2-4

50014 Zaragoza

* Distribuidor oficial autorizado

* Cursos de formación Microinformática Robótica. 8/16 Bits.

Lo que hay que saber...

* ¿Qué capacidad máxima tiene el programa AMSFILE?

— En el folleto de intrucciones se explican los máximos datos que puede manejar dicho programa. Los datos correctos son:
9.999 fichas en total en un fichero.
60 caracteres como máximo en un campo.
20 campos en una ficha.
1.200 caracteres en total en una ficha.

Esto significa que las máximas capacidades que el programa puede usar internamente son ésas. Pero no quiere decir que se puedan usar todas las capacidades a la vez, debido a las limitaciones de los medios de almacenamiento (léase discos). Es decir, según lo anterior, la longitud máxima que podría tener un fichero manejado por AMSFILE sería:

9.999 fichas × 1.200 caracteres = 11.998.800 número total de caracteres.

O lo que es lo mismo 11.717Ks, o alrededor de 11 Megabytes. Como se puede ver estas capacidades desbordan por completo la capacidad que poseen actualmente los Amstrad de la familia CPC, e incluso, la del nuevo PCW8256. Para manejar esa cantidad de datos se necesitaría un disco duro, lo que, según nuestras noticias, todavía no está disponible para los ordenadores Amstrad.

Para hallar la longitud máxima que puede tener un fichero concreto se sigue el método siguiente: Dividir la capacidad libre que tenemos en un disco entre el número de caracteres de la ficha que hemos definido. El resultado es el número máximo de fichas.

La capacidad libre son 169Ks con formato sistema (que exige Amsfile). Además de esto hay que tener en cuenta si se trabaja con dos unidades de disco o con una, ya que si se da esta última condición, hay que descontar lo que ocupan los programas de Amsfile (en conjunto, 43Ks), ya que éstos deben ser siempre accesibles (con dos unidades se puede dejar una unidad entera para el fichero —el drive B— y tener el programa en la otra). A ambas capacidades hay que descontar un máximo de 3Ks, que ocupan los ficheros auxiliares de definiciones y de número de datos. Resumiendo, con dos unidades tenemos libres 166Ks (169-3) y con una, 123Ks (169-43-3).

P.e. El máximo número de fichas que se puede manejar con una ficha de 100 caracteres es la siguiente:

— Una unidad: 123*1024/100 ≃ 1259 fichas.
— Dos unidades: 166*1024/100 ≃ 1700 fichas.

* ¿Qué tengo que hacer para tener una copia de Amsfile en disco?

Para obtener una copia de AMSFILE en disco, el proceso más rápido es éste:

1. Hacer una copia del disco-regalo con ayuda de los comandos 'COPYDISC' (si dispone de 2 unidades de disco) o 'DISCCOPY' (si sólo dispone de una), para el 664 o el comando 'DISCKIT3' para el 6128.

Para ello, introduzca el disco de CP/M en la unidad A; teclee ICPM y pulse ENTER.

Una vez que aparezca la divisa de CP/M, A>, teclee COPYDISC, DISCCOPY o DISCKIT3 (según sea el número de unidades y la versión de CP/M) y proceda como le indica la pantalla.

2. Una vez terminado este proceso, dispóngase a borrar todos los programas que no le son necesarios: para verlos teclee DIR, habiendo introducido previamente la copia realizada en la unidad A.

De los programas que aparecen en pantalla, debe borrar todos EXCEPTO: BRUN.COM, AMSFILE.COM, AMSFILE2.COM, RGOAD.COM.

Para borrar, teclee: ERA nombre-de-programa.extensión.

Por ejemplo, ERA EAMSWORD,BAS [ENTER]
o ERA *, BIN [ENTER]

El otro método es hacer una copia de sólo los ficheros terminados en la extensión .COM, haciendo uso del programa FILECOPY o PIP, según la versión de CP/M.

Una vez hecho esto, hay que ejecutar los programas SYSGEN y BOOTGEN de CP/M 2.2, en los que será el disco fuente (source) el original de regalo y el destino (destination) la copia realizada. La finalidad de estos programas es copiar el sistema operativo en las dos primeras pistas y generar el programa de arranque.

* ¿Qué significa la pregunta, en la opción 'Salidas por Impresora', INTRODUZCA CODIGOS DE IMPRESION...?

Esto se refiere a los códigos de control de la impresora que se envían para que ésta tome determinadas acciones, generalmente se utilizan para poder escribir en distintos tipos de letra. Por ejemplo, para listados de anchura superior a 80 caracteres, quizá se desee mandar el código correspondiente a escritura comprimida, que permite escribir hasta 132 caracteres (generalmente el 15). O listados en letra de alta calidad (normalmente 27, 120, 1), etc. Para mayor información, consulte el manual de su impresora.

Si no se desea ningún tipo de letra especial, simplemente pulse ENTER, y volverá al menú 'Salidas por Impresora'. Si desea mandar algún código de control a la impresora, debe comprobar la secuencia correcta en su impresora, teniendo en cuenta que éstos pueden ser de uno a tres caracteres y que la última vez debe pulsar ENTER, sin haber tecleado nada. Veamos dos ejemplos.

Supongamos que desea mandar el código de comprimido, y que éste corresponde en su impresora al 15 (decimal). Teclearía:

15 [ENTER] [ENTER]

Si deseara mandar los códigos de alta calidad, suponiendo que su impresora tuviera esa posibilidad y que correspondieran A ESC x + 1, lo que debería teclear sería 27 (código que corresponde a ESC), 120 (código de la 'x'), seguido de 1. Es decir:

27 [ENTER] 120 [ENTER] 1 [ENTER] [ENTER]

Observe que para salir siempre se pulsa ENTER sin haber tecleado ningún carácter, lo que le indica al programa que queremos abandonar la opción.

M. A. BARRIOS

AMSTRAD SOFT



P.V.P. (CASSETTE) 1.900;- / (DISCO) 2.900;-



P.V.P. (CASSETTE) 1.900;- / (DISCO) 2.900;-



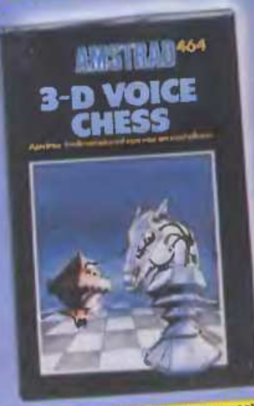
P.V.P. (CASSETTE) 2.500;- / (DISCO) 3.300;-



P.V.P. (CASSETTE) 2.100;- / (DISCO) 3.100;-



P.V.P. (CASSETTE) 2.100;- / (DISCO) 3.100;-



P.V.P. (CASSETTE) 2.300;- / (DISCO) 3.300;-



P.V.P. (CASSETTE) 2.100;- / (DISCO) 3.100;-



P.V.P. (CASSETTE) 1.900;- / (DISCO) 2.900;-



P.V.P. (CASSETTE) 1.900;- / (DISCO) 2.900;-



P.V.P. (CASSETTE) 1.900;- / (DISCO) 2.900;-

ACE DISTRIBUCION
Actividades Comerciales Electrónicas, S.A.
Tarragona, 112 Tel. 325 15 12* Telex 93133 ACEE E 08015 Barcelona

YA DISPONIBLE EN



Y EN TODAS LAS TIENDAS ESPECIALIZADAS

disseny: TF


RPA®
Systems inc.
Multibase 3
Facturación
Gestión de Empresa
Nóminas
Contabilidad
Agenda Robot
RPA
TODAVIA
MAS FACIL
Programas para AMSTRAD 8256/6128/664/464
RPA Systems Inc. te ofrece una amplia gama de programas, hasta 34, para que tu gestión sea más rápida y eficaz.
Programas muy fáciles de usar, con un lenguaje compilado de alto nivel y continuas ayudas en pantalla.
Por eso, si eres pequeño empresario, comerciante o profesional liberal, ahora lo tienes "todavia más fácil".
De venta en los principales almacenes y en tiendas especializadas.
Pidenos información sin compromiso, tu primera sorpresa será su precio.
RPA
Distribuidor exclusivo en España BABETA S.A.
Galileo. 25. Entreplanta A. Tels. 447 97 51 / 98 09. 28015 Madrid.
Distribuidor exclusivo en Catalunya: ACE DISTRIBUCION, S.A.
Tarragona, 112. Tel. (93) 325 15 12. 08015 Barcelona
Telex: 93133 ACEE E
RPA
BASE DE DATOS
664/6128
CUP